



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH A TVORBA NOVÉ E-COMMERCE PLATFORMY

DESIGN AND CREATION OF A NEW E-COMMERCE PLATFORM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Hladík

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Luhan, Ph.D., MSc

BRNO 2019

Zadání diplomové práce

Ústav: Ústav informatiky
Student: **Bc. Petr Hladík**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Informační management
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**
Akademický rok: 2018/19

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Návrh a tvorba nové e-commerce platformy

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem práce je navrhnout a vytvořit novou e-commerce platformu na úrovni prototypu využitelného jako elektronický obchod konkrétního podniku dle zadaných požadavků.

Základní literární prameny:

HOPKINS, C. a O. BAŠE. PHP okamžitě. 1. vyd. Brno: Computer Press, 2014. 134 s. ISBN 978-80-251-4196-0.

LAUDON, K. C. and C. G. TRAVER. E-Commerce 2016: Business, Technology, Society. 12th ed. New Jersey: Pearson, 2016. 912 p. ISBN 978-0-133-93895-1.

SUCHÁNEK, P. E-commerce: elektronické podnikání a koncepce elektronického obchodování. 1. vyd. Praha: Ekopress, 2012. 144 s. ISBN 978-80-86929-84-2.

TARR, A. PHP and MySQL: 24-Hour Trainer. 1st ed. Indianapolis: John Wiley & Sons, 2012. 504 p.
ISBN 978-1-118-06688-1.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně dne 28.2.2019

L. S.

doc. RNDr. Bedřich Půža, CSc.
ředitel

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Hlavní úlohou této diplomové práce je vytvoření prototypu e-commerce platformy, která bude v budoucnu tvořit základ plnohodnotného řešení pro konkrétního živnostníka. Práce se zabývá analýzou současného stavu, analýzou dostupných řešení, popisem vybraných technologií, včetně popisu, jak tyto technologie byly v projektu konkrétně implementovány. Výsledkem práce je funkční prototyp e-commerce platformy.

Abstract

The thesis focuses on developing prototype of e-commerce platform. This platform will be used as a base for a full-fledged e-commerce solution of specific trader in the future. The thesis deals with the analysis of the current state, analysis of available solutions, description of selected technologies, including a description of how these technologies were specifically implemented in the project. The result of this thesis is a functional prototype of e-commerce platform.

Klíčová slova

E-commerce, E-shop, PHP, OOP, Nette Framework, MySQL, Doctrine, Elasticsearch, Composer, RabbitMQ

Keywords

E-commerce, E-shop, PHP, OOP, Nette Framework, MySQL, Doctrine, Elasticsearch, Composer, RabbitMQ

Bibliografická citace

HLADÍK, Petr. *Návrh a tvorba nové e-commerce platformy*. Brno, 2019 [cit. 2019-05-07]. Dostupné z: <https://www.vutbr.cz/studenti/zav-prace/detail/119854>. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jan Luhan.

Čestné prohlášení

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

.....
Petr Hladík
7. května 2019

Poděkování

Tímto bych chtěl poděkovat všem, kteří mi pomáhali při zpracovávání této práce, hlavně těm, kteří mi poskytovali cenné rady a informace, zejména vedoucímu práce Ing. Janu Luhanovi, Ph.D., MSc.

Obsah

1	Úvod	12
2	Teoretická východiska práce	14
2.1	Internet	14
2.1.1	Vznik internetu	14
2.1.2	Služby na internetu	15
2.1.3	Webový prohlížeč	16
2.1.4	Webová stránka	17
2.1.5	Hypertext	17
2.2	E-business	17
2.2.1	E-commerce	18
2.2.2	E-shop	18
2.2.3	Modely e-commerce	19
2.2.4	Vývoj e-commerce	21
2.2.5	E-commerce v České republice	21
2.3	E-commerce z pohledu obchodníků	22
2.3.1	Založení a provoz e-shopu	22
2.3.2	Online marketing	24
2.4	E-commerce z pohledu zákazníků	25
2.4.1	Výhody a omezení	25
2.4.2	Důvěryhodnost e-shopu	26
2.5	Hypertext Markup Language – HTML	27
2.6	CSS	28
2.6.1	CSS Preprocesor	28
2.7	Hypertext Preprocesor – PHP	28
2.7.1	Interpretace jazyka	29
2.7.2	Možnosti jazyka	29
2.8	JavaScript	29
2.8.1	Asynchronous JavaScript and XML – AJAX	30
2.9	Objektové programování	30
2.9.1	Objekt	31
2.9.2	Atributy	31
2.9.3	Metody	31
2.9.4	Třída	31
2.10	Framework	31

2.11	MVC architektura	32
2.11.1	Model	33
2.11.2	View	33
2.11.3	Controller	33
2.12	MVP architektura	33
2.13	Databáze	34
2.13.1	Relační databáze	34
2.14	Správa verzí	35
2.14.1	Lokální systémy správy verzí	35
2.14.2	Centralizované systémy správy verzí	36
2.14.3	Distribuované systémy správy verzí	36
2.15	Composer	37
2.15.1	Satis	37
2.16	Nette Framework	38
2.17	Nittro	38
2.18	Doctrine 2	39
2.18.1	Entity	39
2.19	Elasticsearch	40
2.20	Message broker	40
2.20.1	RabbitMQ	41
2.21	Virtualizace	41
2.21.1	Vagrant	41
2.21.2	Laravel Homestead	42
2.22	Analýza statického kódu	42
2.22.1	Cyklomatická složitost	43
3	Analýza současné situace	44
3.1	Základní informace o živnostníkovi	44
3.2	Značka HanaBrand	45
3.3	Nabízené produkty a služby	46
3.4	Obchodní vztahy	47
3.4.1	B2C	47
3.4.2	B2B	47
3.5	Současná technická situace	47
3.5.1	Technické požadavky současného e-shopu	48
3.5.2	Entity v aplikaci	48
3.5.3	Administrované entity v aplikaci	49
3.5.4	Implementované funkcionality	54
3.6	Statická analýza kódu	55
3.7	Zhodnocení současného stavu	57
3.8	Požadavky na nové řešení	58
3.9	Průzkum dostupných řešení	59
3.9.1	Vlastní řešení	59
3.9.2	Open-source řešení	60
3.9.3	Řešení na míru	61

3.9.4	Pronájem	62
3.9.5	Výsledek průzkumu	63
4	Návrh řešení	64
4.1	Výběr technologií	64
4.1.1	Technologická platforma	64
4.1.2	Laravel Homestead	65
4.1.3	Composer	65
4.1.4	Satis	66
4.1.5	Nette Framework	66
4.1.6	Nittro	67
4.1.7	Doctrine 2	67
4.1.8	Elasticsearch	67
4.1.9	RabbitMQ	68
4.2	Struktura aplikace	68
4.2.1	Části systému	68
4.2.2	Moduly	71
4.2.3	Adresářová struktura	73
4.3	Modelová vrstva	74
4.3.1	Entity	74
4.3.2	Repositáře	77
4.3.3	Kontejnery	77
4.3.4	Fasády	77
4.4	Funkcionality systému	78
4.4.1	Systémová oprávnění	78
4.4.2	Asynchronní zpracování obrázků	78
4.4.3	Automatické sestavování formulářů	80
4.4.4	Export entit do Elasticsearch	81
4.4.5	Import entit z Elasticsearch	82
4.4.6	Lokalizace	82
4.4.7	Vytváření URL adres	84
4.4.8	Administrační rozhraní	85
4.4.9	Uživatelské parametry	88
4.4.10	Varianty produktů	89
4.4.11	Customizace systému	90
4.4.12	Přepisování služeb	90
4.4.13	Přepisování šablon	92
4.4.14	Moduly	93
4.5	Redesign webu	93
4.5.1	Úvodní stránka	93
4.5.2	Detail kategorie	98
4.5.3	Detail produktu	99
4.6	Zhodnocení realizace	101
4.6.1	Statická analýza kódu	101
4.6.2	Plánovaný rozvoj	103

4.6.3	Finanční zhodnocení	104
5	Závěr	106
	Literatura	107

Kapitola 1

Úvod

Internet je jednoznačně nejvýznamnější fenomén současnosti. Ovlivnil většinu lidí, zasáhl do všech sociálních skupin bez ohledu na věk nebo na vzdělání a zcela změnil celá průmyslová odvětví. V rámci internetu lze realizovat celá řada činností, mimo jiné i nakupování. To je čím dál oblíbenější a obrat české e-commerce každý rok strmě stoupá. Po technické stránce je pro společnost působící v oblasti elektronického obchodování velice důležité mít správně sestavenou platformu, skrz kterou realizuje svoji obchodní činnost.

Tato práce se zabývá návrhem a vytvořením prototypu e-commerce platformy, která bude v budoucnu tvořit základ plnohodnotného řešení pro konkrétního živnostníka, paní Hanu Hladíkovou, provozovatelkou internetového obchodu www.hanabrand.cz.

Druhá kapitola obsahuje shrnutí základních pojmů a informací potřebné k pochopení dalších částí práce a k pochopení problematiky týkající se elektronického obchodování. Jsou přiblížena i témata, na která se v dalších částech práce nepřímo odkazuje, ale jsou podstatná pro pochopení ostatních teoretických základů nebo pro vytvoření uceleného pohledu na danou problematiku.

Ve třetí kapitole jsou uvedeny podstatné informace o živnostníkovi, pro kterého bude vytvořena e-commerce platforma. Jsou uvedeny informace, jak o podnikatelské činnosti živnostníka, tak analýza současného řešení. V závěru této kapitoly se nachází analýza dostupných řešení na základě konkrétních požadavků živnostníka, včetně výsledku této analýzy, která se stala rozhodujícím faktorem při realizaci této práce.

Čtvrtá kapitola popisuje návrhovou a implementační část vytvoření e-commerce platformy na úrovni prototypu. Uvádí, jaké technologie byly při návrhu zvoleny, včetně popisu, jak konkrétně byly využity, aby tento prototyp tvořil kvalitní technologický základ pro budoucí vývoj v plnohodnotnou e-commerce platformu. V závěru této kapitoly se nachází i zhodnocení celé realizace včetně finančního ohodnocení.

Záměrem celého projektu vývoje této e-commerce platformy je vytvoření řešení, které bude splňovat veškeré živnostníkové požadavky, nahradí jeho současně používané řešení a platforma se bude dále rozvíjet.

Kapitola 2

Teoretická východiska práce

V této části jsou shrnuty základní pojmy a informace, které jsou klíčové pro pochopení dalších částí práce a k pochopení problematiky v oblasti e-commerce. Jsou zde vysvětleny i témata, na která se v dalších částech práce přímo neodkazuje, ale jsou podstatná pro pochopení ostatních teoretických základů nebo pro vytvoření uceleného pohledu na danou problematiku.

2.1 Internet

Internet je celosvětová síť (desítky tisíc jednotlivých sítí – možná i mnohem více) propojující všechny kontinenty, státy a města tak, že jakýkoliv počítač může kdykoliv komunikovat s jakýmkoliv jiným počítačem, ať je kdekoliv na planetě. Internet nikdo neřídí, funguje díky tomu, že všichni lidé mají společný zájem – chtějí být propojeni [1].

2.1.1 Vznik internetu

První předchůdce internetu (Arpanet), vznikl v roce 1969. Vytvořila ho instituce Advanced Research Project Agency (ARPA). Jeho vznik zaštiťovalo Ministerstvo obrany USA. Tato síť byla z počátku použita pouze pro vědecké účely, následně z ní vznikly dvě samostatné sítě – Arpanet a Milnet (armádní síť). V roce 1981 vznikla síť Bitnet, která propojovala americké vysoké a střední školy. Avšak v těchto sítích

vznikl problém komunikace na různých platformách. Proto po intenzivních výzkumech vznikl protokol TCP/IP, který je používán dodnes. Samotný vznik internetu v podobě, jak ho známe dnes, je datován od roku 1993, kdy Tom Berners Lee vymyslel nový způsob výměny informací, ten je známý pod zkratkou WWW (World Wide Web) [2].

Internet se v České republice začal rozšiřovat až na přelomu let 1995 a 1996, kdy na trh vstoupili komerční poskytovatelé internetového připojení. I přes to, že v těchto letech byla Česká republika připojena k internetu již 3 roky, ke komerčnímu používání internetu došlo až po pádu společnosti Eurotel, jejíž monopol se vztahoval mimo jiné i na veřejné služby přenosu dat [2].

2.1.2 Služby na internetu

V rámci internetu lze využít celá řada služeb. Zde jsou vysvětleny ty, které jsou hojně využívané pro komerční využití [3].

World Wide Web (WWW)

Označení pro vzájemně provázený multimediální obsah poskytovaný pomocí protokolu HTTP. Uživatelé obsah konzumují pomocí webových prohlížečů [4].

E-mail

Zasílání textových zpráv mezi jednotlivými uživateli internetu je k dispozici již od raných dob internetu. Dnes je zajišťován pomocí protokolu SMTP, který byl vyvinut v 70. letech [4].

Díky své rychlosti, nízkým nákladům, pohodlí a snadnému archivování je dnes pro firmy základním komunikačním nástrojem. Lze mimo jiné možné například využít k potvrzení přijaté objednávky z internetového obchodu [3].

Instant masseging (IM)

Instant messaging je technologie, která formou aplikace umožňuje uživatelům sledovat, kteří uživatelé jsou připojeni k síti a dle potřeby s těmito uživateli komunikovat, například zasláním zprávy (chatovat) nebo posílat soubory. Umožňuje tedy komunikovat přes síť mezi uživateli v reálném čase [5].

Některé aplikace dovolují uživatelům i použití webových kamer a mikrofونů a díky tomu je služba ještě více populární. Pro některé uživatele je tato technologie přijatelnější, než okamžitá komunikace, protože ji vnímají méně dotěrnější, než například komunikace přes telefon [5].

Telefonování přes internet

Díky rozšíření internetu do firem, je možné telefonovat přes internet. Postupně tak vznikly 3 základní možnosti telefonování s využitím internetu – telefonování z počítače na počítač, z počítače na telefon a telefonování přes internet prostřednictvím mobilního telefonu. Hlavní výhodou telefonování přes internet je snížení nákladů na volání [3].

Sociální služby

„Sociální síť je internetová služba, která registrovaným uživatelům umožňuje komunikovat mezi sebou. Komunikace může být veřejná nebo soukromá, je zpravidla založena na sdílení textových příspěvků, fotografií a videa“ [6].

2.1.3 Webový prohlížeč

Webový prohlížeč je program v počítači, který dokáže zobrazit webovou stránku. Existuje jich několik, např. Internet Explorer, Mozilla Firefox, Chrome, Safari, Opera a další [7].

2.1.4 Webová stránka

Webová stránka je označení dokumentu s informacemi, který je dostupný pomocí internetu a webového prohlížeče a pro návštěvníka webové stránky je prezentován estetickým a čitelným způsobem [8].

Webové stránky se používají pro různé účely, jak pro individuální použití, tak firemní a mohou být buď statické, nebo dynamicky generované [9].

Statická webová stránka je stránka, kde je veškerý obsah pevně kódován do samotné stránky. Tyto stránky jsou obvykle kódované pouze za použití HTML (HyperText Markup Language). Dynamické webové stránky jsou generovány skriptem, který sám sestavuje HTML dokument. Tento dokument může vygenerovat například za použití dat z databáze. Dynamické webové stránky lze vytvářet za použití několika programovacích jazyků, včetně PHP [9].

2.1.5 Hypertext

„Text, ve kterém fungují některé výrazy jako odkazy. Prostřednictvím těchto odkazů se přenesete na jinou část dokumentu nebo na úplně nový dokument“ [10].

2.2 E-business

„Elektronické podnikání (e-business) je oborem, který zaznamenal velmi rychlý vývoj. Zpočátku byl e-business chápán pouze jako internetové obchody, rezervační systémy apod., tedy aktivity, které jsou dnes označovány jako elektronické obchodování (e-commerce). E-business má širší význam a patří sem i mnoho dalších aktivit, jejichž cílem je zejména podpora a zvýšení efektivity podnikových interních a externích procesů. Na stejnou úroveň jako firmy se v rámci e-business řadí i běžní uživatelé, kteří jsou na jedné straně klíčovou cílovou skupinou z hlediska elektronicky realizovaných obchodů, na druhé straně využívají Internet a jeho služby pro účely vzdělávání, zábavy nebo například pro pracovní účely“ [11].



Obrázek 2.1: Části elektronického podnikání, zdroj: vlastní zpracování dle [11]

2.2.1 E-commerce

E-commerce je široký pojem, který se používá k označení veškerých obchodních transakcí realizovaných pomocí internetu a elektronických prostředků. E-commerce patří do rodiny e-business (elektronického podnikání) [12].

Pro e-commerce jsou důležité digitální obchodní transakce mezi organizacemi a mezi jednotlivci. Digitální transakce zahrnují všechny transakce zprostředkované prostřednictvím digitálních technologií. Z velké části to jsou transakce, ke kterým dochází přes internet. Obchodní transakce zahrnují výměnu hodnoty (např. peněz) přes organizační nebo individuální hranice za produkty a služby. Výměna hodnoty je důležitá pro pochopení hranic elektronického obchodu – bez výměny hodnoty nedochází k žádnému obchodu [13].

2.2.2 E-shop

E-shop je elektronický internetový obchod v podobě speciální webové aplikace, která slouží ke zprostředkování obchodních transakcí prostřednictvím internetu [14].

Základem je katalog výrobků, který je obvykle řazený do kategorií. V těchto kategoriích se pak nacházejí jednotlivé produkty, které jsou nabízeny zákazníkovi k zakoupení. Ze zákona musí být na e-shopu také další informace, jako jsou například kontaktní údaje, obchodní podmínky apod. Poté, co si uživatel vybere svůj produkt, dochází k nákupnímu procesu. Prvním krokem je přidání produktu do tzv. košíku (elektronická podoba klasického nákupního košíku), zde se shromažďují vybrané produkty. Poté uživatel vyplní doručovací a fakturační údaje (v některých případech lze využít uživatelský účet a uživatelské údaje jsou doplněny automaticky). Propracovanější e-shopy dolují i například odložit nákup na později nebo zaplatit rovnou pomocí platební karty [14].

2.2.3 Modely e-commerce

„Existují čtyři základní modely elektronického obchodování, které využívají prostředí internetu a možnosti moderních informačních a komunikačních technologií. Modely jsou definovány pomocí směru marketingové komunikace – od koho vychází a komu je určena“ [15].

„Patří sem i většina činností spadajících pod elektronický marketing, např. online reklama, email marketing, nejružnější affiliate programy a všechny aktivity na podporu internetového obchodování. V širším pojetí mohou do pojmu e-commerce patřit i jakékoliv webové stránky, které nabízí konkrétní produkty či služby a umožňují i jejich objednávku například emailem“ [15].

B2B (Business to Business)

B2B označuje obchodní vztah mezi dvěma podniky. Tento obchodní vztah se odehrává zpravidla prostřednictvím internetu. Typické pro řešení tohoto obchodního vztahu je elektronická výměna dat, například fakturace, řízení skladů, online bankovníctví, online katalogy, online aukce a propojování podnikových informačních systémů [16].

Týká se firem, které chtějí nakupovat (například materiál nebo zboží) z důvodu zisku. Poptávka je ovlivněna množstvím zakázek, které má jeden z partnerů [17].

Tento obchodní vztah nemá přímý vliv na konečného spotřebitele. Nejdůležitějším úkonem v B2B marketingu patří budování důvěry [18].

B2C (Business to Customer)

B2B je označení pro model elektronického podnikání, který se zaměřuje na koncové klienty. Protikladem je B2C business. Iniciativa k obchodu vychází směrem od podniku a cílovou skupinou jsou zákazníci (spotřebitelé). Obchodní vztah se odehrává prostřednictvím internetu. Internetové stránky zprostředkovatelů jsou zákazníkům snadno dostupné a mohou nabízet zákazníkům informace a jednoduchý způsob nákupu. Internet umožňuje zákazníkům porovnávat nabídky různých prodejců [19].

Samotná kategorie je rozdělena do tří kategorií. Do první kategorie spadá informování potenciálního zákazníka o produktech. Druhou kategorií je získávání zpětné vazby skrz různé formuláře. Třetí a největší kategorií je uzavření celého obchodu s možností objednání a zaplacení zboží [18].

B2C je zřejmě nejrozšířenější forma internetového podnikání. Jedná se o přímý prodej a podporu zákazníkovi [17].

C2B (Customer to Business)

C2B je model elektronického podnikání, kdy iniciativa pochází od zákazníků (spotřebitelů) směrem k firmám. Zákazníci kontaktují firmy a vznášejí dotazy, podněty a náměty, například pro inovaci výrobků nebo služeb firmy. Takový model lze podpořit například zřízením diskuzního fóra nebo účet na sociálních sítích [20].

C2C (Customer to Customer)

C2C je druh elektronického obchodování, ve kterém probíhá obchodní vztah mezi dvěma spotřebiteli a to většinou prostřednictvím třetí strany. Iniciativa přitom vychází od jednoho spotřebitele směrem k jinému spotřebiteli. Příkladem takovýchto obchodních vztahů jsou například online burzy, aukce a výměny zboží [21].

2.2.4 Vývoj e-commerce

„Nakupování online je globální a rapidně rostoucí fenomén. Na světě žije téměř 7,5 mld. lidí, z toho 3,5 mld. lidí používá internet a přes 1,5 miliardy lidí nakupuje online“ [22].

Nakupování zboží a služeb přes internet se stává samozřejmostí. Svědčí o tom neustále rostoucí počet nákupů online a díky tomu i stále se zvyšující tržby online obchodů. Vedle klasických e-shopů se rozvíjí i další internetové služby související s internetovým prodejem, jako jsou například zásilkovny (uloženky), kde je možné si objednané zboží vyzvednout. Dále se rozvíjí i rozvážkové služby, například rozvoz jídel nebo potravin. Spolu s internetovým prodejem se rozvíjí i srovnávače cen, agregátory nabídek zboží nebo služeb a další [22].

Pro e-commerce hraje velkou roli sezónnost prodeje s velkým významem vánoc – obvykle e-shopy utrží za poslední 4 měsíce přibližně stejně, jako za předchozích 8 měsíců. V poslední době jsou ale sezónní rozdíly vyrovnávány nákupy zboží denní potřeby. Nejslabšími měsíci v tržbách e-shopů jsou většinou letní měsíce [22].

„Podle serveru eMarketer se maloobchodní globální online tržby přiblížily v roce 2016 úrovni 1,9 bilionu dolarů a do roku 2020 by se měly zdvojnásobit na 4 bil. dolarů. I tak se e-commerce tržby budou podílet na celkových maloobchodních tržbách jen 14,6 %. Největším světovým online trhem je Čína, jejíž podíl poroste i do budoucna. Následuje online trh v USA“ [22].

2.2.5 E-commerce v České republice

Dlouhodobě roste počet Čechů s přístupem k internetu. V roce 2016 se počet uživatelů internetu v České republice oproti roku 2010 zvýšil zhruba o 15 % a díky tomu se Česká republika počtem uživatelů internetu přiblížila průměru v Evropské unii [22].

„Mezi ekonomicky aktivními, vysokoškolsky vzdělanými a mladými do 30 let užívají dnes internet už téměř všichni“ [22].

V roce 2016 mělo 76,1 % domácností v České republice přístup k internetu. Také roste obliba v nakupování přes internet. V roce 2007 podle průzkumu ČSÚ za poslední

rok nakoupilo přes internet 15 % obyvatel starších 16 let. V roce 2010 jich nakoupilo již 25 % a v roce 2017 již 43 % [22].

2.3 E-commerce z pohledu obchodníků

Hlavní výhodou pro provozovatele e-shopů je široká působnost a v podstatě neomezená otevírací doba. E-shop může s přehledem prodávat po celé zemi (a v mnoha případech i za hranicemi). Internetový obchod tedy nemusí být limitován žádným konkrétním místem. Široká působnost se pak pozitivně odráží v marketingu, kdy umožňuje širší zásah při nižších nákladech na propagaci. Na internetu lze snadno měřit efektivitu reklamy a konfigurovat ji v závislosti na těchto výsledcích a potřebách zákazníka [23].

Na e-shop lze uvést o produktu mnoho informací a nabídnout i komponenty a substituty konkrétního produktu. Zároveň mají v porovnání s kamennými prodejnami i nižší náklady na provoz. Ale i zde platí, že čím více se bude byznys rozšiřovat, tím se budou i náklady na provoz přibližovat k nákladům za provoz kamenné prodejny [23].

Jako určitou nevýhodu lze na internetu označit ceny produktů. Díky cenovým srovnávačům může zákazník jednoduše vidět cenové rozdíly s konkurencí. Pokud by chtěl srovnávat ceny zboží v kamenných prodejnách, stálo by to zákazníka mnohem více úsilí a času, proto si lze v kamenných prodejnách dovolit vyšší cenu produktu oproti konkurenci [23].

2.3.1 Založení a provoz e-shopu

Základní kroky, které jsou potřeba podniknout k založení a úspěšnému provozování e-shopu, kterými by se měli podnikatelé zakládající e-shop zabývat:

- vypracovat podnikatelský plán (včetně cílů, strategií a financování),
- právní ošetření podnikání, registrace na příslušných úřadech,

- výběr technického řešení e-shopu (open-source, programování na zakázku, pronájem),
- výběr a nákup vhodné domény a poskytovatele hostingových služeb,
- výběr vhodných dodavatelů (výrobků i spotřebního materiálu),
- zajištění vhodné formy distribuce,
- promyslet koncept e-shopu (struktura, funkce, bezpečnost, dostupnost),
- vytvoření vzhledu a účelného obsahu (copyright, SEO, obchodní podmínky) [24].

Open-source

Největším lákadlem řešení open-source je jeho cena, protože toto řešení je zdarma. Zdarma jsou základní moduly, které vystačí na běžné potřeby e-shopu. Zdarma mohou být i další rozšíření a pluginy. Další výhodou je otevřený kód, který lze libovolně modifikovat a rozvíjet dle vlastních požadavků. Open-source e-commerce řešení tedy využijí ti, kteří chtějí levný, jednoduchý e-shop a nebudou požadovat víc, než jsou jeho základní funkce. Dále toto řešení využijí ti, kteří chtějí použít otestované řešení jako solidní základ pro vlastní platformu, kterou budou dále chtít sami rozvíjet [25].

Open-source ale neznamená e-shop zdarma nebo nulové náklady. I správa svobodného softwaru bude potřebovat zkušeného vývojáře, který se dokáže orientovat v cizím kódu, zná architekturu aplikace a aby požadovanou úpravu provedl dobře a nezanesl do aplikace další chyby [25].

Bohužel svobodný software v sobě ukrývá i své nevýhody – kolísající kvalita kódu a dostupnost podpory. Díky tomu, že lze do kódu libovolně zasahovat, žádné open-source řešení nezaručí, že bude fungovat bez chyb a nečekaných pádů. Veřejně přístupný kód také přitahuje pozornost útočníků, kteří vyhledávají bezpečnostní chyby [25].

E-shop na míru

E-shop na míru má tu výhodu, že přesně odpovídá požadavkům zadavatele. Na počátku může zadavatel přesně specifikovat, jakou funkcionalitu požaduje. Tím se například liší od krabicového řešení, protože vznikne systém, který přesně splňuje potřeby zadavatele [26].

E-shop na míru je ale poměrně drahá varianta k vytvoření vlastního e-shopu. Většinou se tak spíše vyplatí u velkých společností, které mají dostatek peněz na takovouto investici [26].

„E-shop na míru je pro podnikatele ta nejlepší varianta pod podmínkou, že na ni má“ [27].

Pronájem e-shopu

Pronájmu e-shopu se také říká krabicové řešení e-shopu. Lze s ním začít podnikat téměř okamžitě, pronajímají se za danou měsíční částku. Krabicově řešené e-shopy jsou již hotové a funkční. Stačí je jen nastavit a naplnit produkty. Mezi hlavní výhody patří minimální starosti s e-shopem, bezpečnost, spolehlivost a dopředu stanovená cena pronájmu. Nevýhodami tohoto řešení je menší možnost úprav a fakt, že e-shop nepatří nájemníkovi. Je tedy vhodnou formou k vytvoření e-shopu spíše začínajícím podnikatelům [27].

2.3.2 Online marketing

Online marketing je označení pro propagaci produktů nebo služeb na internetu [28].

Rozdělení online marketingu:

- inbound marketing – dlouhodobé formování a podpora vztahu s potenciálními zákazníky, například pomocí různého obsahu pomáhá, vzdělává nebo baví (populární obsahový marketing),

- outbound marketing – zaměřuje se naopak na efektivní propagaci produktu nebo služby [28].

Hlavní nástroje online marketingu:

- internetové stránky – jádro prezentace na internetu,
- sociální sítě – oslovování potenciálních zákazníků přes udržované a aktualizované stránky s poutavými příspěvky nebo reklamou,
- e-mail marketing – dlouhodobě úspěšná metoda založená na prohlubování vztahu mezi odesílatelem a příjemcem (potenciálním zákazníkem) například formou newsletterů,
- PPC kampaně – (výkonnostní marketing) umístění textových sdělení nebo bannerů v obsahové nebo vyhledávací síti, čímž si inzerent získává pozornost uživatelů,
- affiliate marketing – získávání nových zákazníků od partnerů pomocí provozního systému,
- SEO – optimalizace pro vyhledávače, optimalizace faktorů ovlivňujících zviditelnění webu ve výsledcích vyhledávání [28].

2.4 E-commerce z pohledu zákazníků

„Internet je jako obchodní místo unikátní hned v několika směrech: např. koncentrace obchodů na jednom místě. Všechny jsou okamžitě dosažitelné, nekoupím-li v prvním, mohu bez námahy pokračovat v objevování další nabídky, jako kdybych jel na druhý konec města nebo dokonce světa. Existuje možnost rychlého porovnání sortimentu a cen. Rychlý přehled konkurence v oboru, jen na základě zadání třeba jednoho klíčového slova“ [29].

2.4.1 Výhody a omezení

Základní výhody e-commerce pro zákazníky:

- nižší ceny produktů a služeb, možnost okamžitého srovnání cen s konkurencí,
- možnost většího výběru produktů a obchodníků,
- větší pohodlí a úspora času – snadná dostupnost k nákupu, možnost nakupovat kdykoli a odkudkoli,
- snadné získání detailních informací o produktech a službách,
- možnost rychlého dodání,
- slevy a výhody pro zákazníky [30].

Základní nevýhody e-commerce pro zákazníky:

- zákazníci si nemohou zboží osahat a ozkoušet,
- nabízené zboží není skladem,
- někteří zákazníci se bojí problémů s reklamací nebo vrácením zboží,
- obava ze zneužití kreditní karty,
- zákazník potřebuje poradit s výběrem zboží a nemá se na koho obrátit [31].

2.4.2 Důvěryhodnost e-shopu

Během pouhých prvních pár vteřin návštěvník internetových stránek zhodnotí, zda je pro něj e-shop důvěryhodný a zajímavý. Navíc je pro zákazníky velice důležité, komu poskytnou osobní údaje a informace o platební kartě. Z toho důvodu je potřeba, aby webové stránky působily důvěryhodně [32].

Doporučované základní úpravy pro obecné zvýšení důvěryhodnosti e-shopu:

- propracovaný design webu, aktualizovaný jednoduchý přehledný a nepříliš agresivní vzhled,
- vizuální kontakt s člověkem nebo maskotem,
- aktuálnost e-shopu,
- zakomponování bezpečnostního symbolu,
- přidání možnosti snadného kontaktu [33].

„Důvěryhodný je takový e-shop, který působí moderně, aktivně, stará se o své zákazníky a umožňuje jim kontakt, kdykoli je potřeba. Zároveň za ním stojí konkrétní sympatičtí lidé s příběhem. Zapracujte tyto úpravy na svůj e-shop a uvidíte pozitivní vliv na obrát i pokles míry okamžitého opuštění stránek. A pokud chcete získat ještě více objednávek, nezapomeňte vytvořit perfektní stránku O nás“ [33].

2.5 Hypertext Markup Language – HTML

HTML je značkovací jazyk pro hypertext. Je jedním z jazyků, které umožňují vytvářet dokumenty, které je pak možné publikovat na internetu. Je definován množinou značek a jejich atributů. Mezi značky se uzavírají části textu a tím se určuje význam obsaženého textu. Jednotlivé značky se uzavírají mezi úhlové závorky „<“ a „>“. Součástí elementu mohou být v obsahu další vnořené elementy. Pomocí atributů se elementu doplňují informace, které definují vlastnosti elementu [34].

Značky neboli „tagy“ mohou být párové a nepárové. Párové značky se uzavírají koncovou značkou, která je shodná s počáteční, akorát se před názvem vpisuje znak lomítka „/“ [34].

2.6 CSS

CSS (Cascading Style Sheets – kaskádové styly) je zápis, který určuje vzhled (barvy, dekorační obrázky, rozmístění prvků) HTML dokumentu [35].

2.6.1 CSS Preprocessor

CSS nabízí možnost jak strukturovat styly uvnitř souboru. Vždy musí být uveden jeden nebo více CSS selektorů, po kterých musí následovat blok s výčtem vlastností, které se mají pro dané selektory aplikovat. Díky tomu mohou vznikat dlouhé řetězce selektorů. CSS také nenabízí možnost využití proměnných, například pro aplikaci hodnot, které se ve stylech opakují. CSS preprocessory tyto a další nedostatky obyčejného CSS eliminují a poskytují mnohem více funkcí, než klasické CSS. Ve výsledku je pak zápis přeložen do klasického CSS tak, aby mohlo být využitelné pro nastýlování HTML dokumentu [36].

Mezi CSS preprocessory patří například LESS nebo Sass [37].

2.7 Hypertext Preprocessor – PHP

„PHP je programovací jazyk, který pracuje na straně serveru. S PHP můžete ukládat a měnit data webových stránek. Původní význam zkratky PHP byl Personal Home Page. Vzniklo v roce 1996, od té doby prošlo velkými změnami a nyní tato zkratka znamená PHP: Hypertext Preprocessor” [38].

Je nejoblíbenějším skriptovacím jazykem na straně serveru v oblasti webového vývoje. Pohání téměř 78,9 % webových stránek. Tento jazyk vyvíjí skupina vývojářů s názvem The PHP Group a ta ho zdarma distribuuje jako volně dostupný skriptovací jazyk. Lze jej zprovoznit téměř na všech operačních systémech a na všech platformách [39].

Jazyk je určen pro zpracování dat tak, aby je bylo možné dynamicky zobrazovat na webových stránkách – provádění matematických výpočtů, převádění formátů

a spolupráce s databázemi. Díky tomu umožňuje transformovat statické stránky tak, aby reagovaly na uživatelské podněty [39].

2.7.1 Interpretace jazyka

PHP je imperativní (procedurální) interpretovaný multiparadigmatický programovací jazyk [40]. Imperativní programovací jazyk je jazyk, který k řešení úlohy tedy využívá algoritmu (postup, jak se má daná úloha vyřešit) [40].

Interpretovaný programovací jazyk je jazyk, který je do strojového kódu překládán až za běhu programu (tím se liší od kompilovaných, kde se před spuštěním musí zdrojový kód překompilovat pomocí kompilátoru do strojového kódu a až potom spustěn). Zdrojový kód se za běhu překládá pomocí interpretu a ten příkazy při překladu zároveň provádí (po částech) [41]. Multiparadigmatický programovací jazyk je jazyk, který podporuje více jak jedno programovací paradigma [40].

Programovacím paradigmatem si lze představit základní programovací styl, ve kterém je program naprogramován. Rozdíl je v pojmech a abstrakcích, které tvoří prvky programu (objekty, funkce, proměnné, omezení atd.). Programovací paradigma rozlišujeme na: aspektově orientované, deklarativní (funkcionální, logické), generické, imperativní, paralelní a strukturované (objektově orientované, rekurzivní) [42].

2.7.2 Možnosti jazyka

PHP má spoustu možností pro práci, dokáže pracovat s databázemi, s velkým množstvím internetových protokolů (např. IMAP, POP3, SMTP atd.), dokáže vytvářet (editovat) soubory a obsahuje spoustu funkcí na zpracování řetězců. PHP je zdarma a volně šiřitelné [41].

2.8 JavaScript

JavaScript je multiplatformní a objektově orientovaný, odlehčený skriptovací jazyk, který se spouští na hostovaném zařízení (např. webový prohlížeč). Může se vázat na

objekty hostovaného prostředí a může tak poskytovat prostředky pro jejich ovládání [43].

Neobsahuje statické typování proměnných, ani silnou typovou kontrolu jako v některých jiných programovacích jazycích [43].

Umožňuje manipulovat s prvky webové stránky pro její oživení. Je tedy základem dynamického webu [44].

2.8.1 Asynchronous JavaScript and XML – AJAX

AJAX (česky „asynchronní JavaScript a XML“) je populární způsob pro tvorbu vysoce interaktivních aplikací. Základní princip je aktualizovat pouze části stránky bez nutnosti načítání a vykreslování celé stránky [45].

„AJAX však dovoluje poslat pomocí JavaScriptu separátní HTTP požadavek a data, která nám server vrátí, pak můžeme v JavaScriptu libovolně dále zpracovávat – např. je pomocí DOM zaintegrovat do právě zobrazované stránky. Lze tak interaktivně aktualizovat pouze část stránky” [45].

2.9 Objektové programování

„Objektově orientované programování (dále jen OOP) nevzniklo náhodou, ale je důsledkem vývoje, který k němu směřoval. Jedná se o moderní metodiku vývoje softwaru, kterou podporuje většina programovacích jazyků. Častou chybou je, že se lidé domnívají, že OOP se využívá pouze při psaní určitého druhu programů a jinak je na škodu. Opak je pravdou – OOP je filosofie, je to nový pohled na funkci programu a komunikaci mezi jeho jednotlivými částmi. Mělo by se používat vždy, ať už děláme malou utilitku nebo složitý databázový systém. OOP není jen technika nebo nějaká doporučená struktura programu, je to hlavně nový způsob myšlení, nový náhled na problémy a nová éra ve vývoji softwaru” [46].

Objektem je myšlen soubor dat a funkcí, čímž je patrný velký rozdíl objektového a strukturovaného programování, kdy strukturované od sebe přísně odděluje data a funkce [47].

2.9.1 Objekt

Je základní jednotkou, která dokáže udržovat svůj stav a integrovat s okolím zasíláním a přijímáním zpráv [48].

2.9.2 Atributy

Atributy jsou vlastnosti neboli data (proměnné), která objekt v sobě uchovává. Někdy se o nich hovoří jako o vnitřních stavech objektu [49].

2.9.3 Metody

Metody jsou schopnosti, které je objekt schopen vykonávat. Mohou mít parametry a mohou vracet hodnotu stejně jako funkce. Metoda se od funkce liší tím, že metoda patří objektu, zatímco funkce nepatří žádnému objektu (to je také problém neobjektového programování) [49].

2.9.4 Třída

Aby bylo možné vytvořit objekt, je nutné nejdříve vytvořit třídu. Třída je vzor, podle kterého se objekty vytvářejí, definuje vlastnosti a schopnosti. Objekt, který se vytvoří podle třídy, se nazývá instance. Instance mají stejné rozhraní jako třída, podle kterého se vytvoří, ale navzájem se liší svými atributy [49].

2.10 Framework

Jedním z hlavních společných cílů při programování by mělo být šetřit si práci. Toho dosahujeme použitím správných návrhů funkcí, metod a tříd, které lze uplatnit na více místech. Tohoto cíle lze také dosáhnout použitím existujících osvědčených a optimalizovaných knihoven a nástrojů [50].

Framework je ucelený soubor tematicky zaměřených knihoven, které ulehčují práci při programování aplikací. To zajišťuje psaní méně kódu, kód je přehlednější a rych-

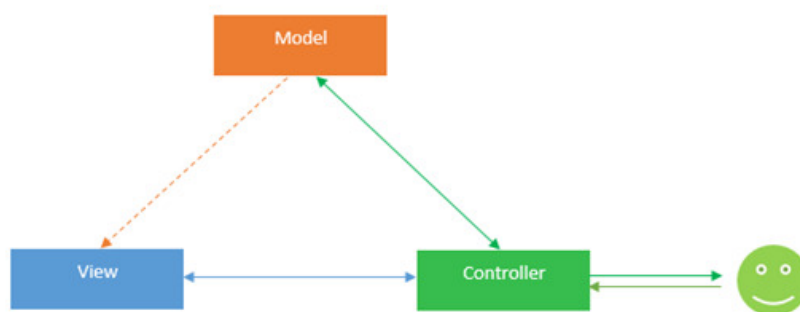
lejší vývoj aplikace. Většina frameworků pracuje s architekturou MVC, která rozděluje aplikaci na nezávislé vrstvy [51].

„Z praktických zkušeností vám mohu s klidem říci, že pokud programujete v čistém PHP, tak je asi 50 % vašeho kódu nepotřebný balast. S použitím kvalitních knihoven napíšete aplikaci za polovinu času a s polovičním počtem řádků. Aplikace bude snadno udržitelná a její tvorba vás bude bavit. Nebudete stále vynalézat kolo a řešit ty samé rutinní záležitosti (jako např. jak ověřit, zda je uživatel administrátor), budete se naopak věnovat vymýšlení zajímavých funkcí té dané aplikace a o tom programování přeci je“ [52].

2.11 MVC architektura

MVC architektura neboli architektura Model-View-Controller, je architektura budování aplikace, která oddělí aplikaci na 3 logické části [34].

„Základní myšlenkou MVC architektury je oddělení logiky od výstupu. Řeší tedy problém tzv. "špagetového kódu", kdy máme v jednom souboru (třídě) logické operace a zároveň renderování výstupu. Soubor tedy obsahuje databázové dotazy, logiku (např. PHP operace) a různě poházené HTML tagy. Vše je zamotané do sebe jako špagety. Kód se samozřejmě špatně udržuje, natož rozšiřuje“ [53].



Obrázek 2.2: Znázornění architektury MVC, zdroj: vlastní zpracování dle [54]

Aplikace se rozdělí na komponenty 3 typů (3 logické části) – Model, View (Pohled) a Controller (nepřekládá se, ale dalo by se přeložit jako „prostředník“) [53].

2.11.1 Model

Model obsahuje logiku aplikace. Mohou to být například výpočty, dotazy na databázi, validace apod. Model neví nic o výstupu. Jeho funkce je přijetí parametrů, zpracování a vydání dat ven. Model neví, jak se k němu data dostala a jak budou dále zpracovaná data interpretována [53].

2.11.2 View

Pohled (View) se stará o interpretaci výstupu aplikace. Šablony lze samozřejmě vkládat do sebe, abychom eliminovali opakování stejného kusu kódu v několika šablonách [53].

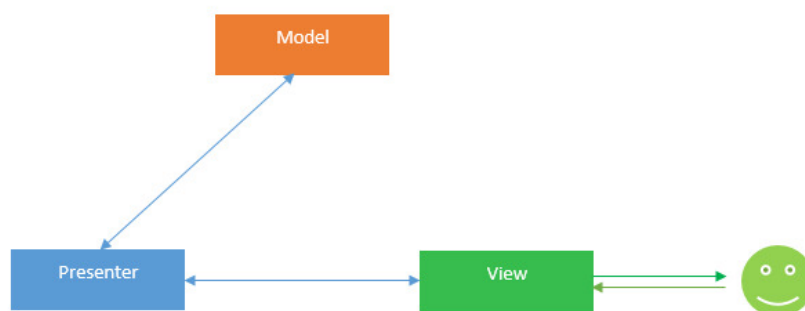
2.11.3 Controller

Controller, lze přeložit jako „řadič“ [55].

Controller zastává funkci prostředníka, se kterým komunikuje uživatel, model i pohled. Drží systém pohromadě a propojuje jednotlivé komponenty [53].

2.12 MVP architektura

MVP (Model-View-Presenter) je návrhový vzor, který vychází ze vzoru MVC. Z něj dědí většinu vlastností, ale hlavním rozdílem je v zodpovědnosti vrstvy Presenter. Controller definuje chování a spravuje typicky několik pohledů (views). Presenter většinou spravuje pouze jediný pohled. V MVC může také pohled komunikovat přímo s modelem, zatímco v MPV má za veškerou „komunikaci“ mezi pohledem a modelem pouze Presenter [56].



Obrázek 2.3: Znázornění architektury MVP, zdroj: vlastní zpracování dle [54]

2.13 Databáze

„Databáze neboli datová základna (*Data Base*) je místo, kam se ukládají určitým způsobem organizované a strukturované údaje. Přístup k údajům uložených v databázi obstarává databázový systém, kterému se říká *SŘBD – Systém Řízení Báze Dat* (anglický termín *DBMS – Database Management System*). Běžně se označením databáze myslí jak uložená data, tak i software“ [57].

Slovem databáze běžně označujeme každý uspořádaný soubor dat [47].

Databáze slouží primárně jako nástroj pro snadnější a efektivnější extrakci informací. Díky uspořádání se z dat stává přehledný systém informací, ze kterého se dají snadno čerpat informace a znalosti [58].

Typy databází lze rozlišovat dle struktur. Mezi základní typy se řadí:

- ploché,
- hierarchické a síťové,
- relační,
- objektové [59].

2.13.1 Relační databáze

Relační databáze jsou v současnosti nejpoužívanějším typem databází [59].

„Součástí relačních databází jsou tabulky, které označujeme jako relace. Jedná se v podstatě o tabulky se sloupci nazývanými atributy, dále pak řádky, které se označují jako záznamy. U každého z atributů je stanoven datový typ a doména. K ukládání samotných informací zde slouží řádky“ [60].

Je doporučeno vytvářet unikátní identifikátory řádků, které jednotlivé záznamy identifikují. Tyto identifikátory se nazývají primární klíče. Dále mohou tabulky obsahovat i cizí klíče. Tyto klíče mají zásadní funkci při návrhu databáze a to, že definují odkaz na primární klíč z jiné tabulky [59].

MySQL

MySQL je populární relační databázový systém, který je dostupný na většině dnešních webhostingů. Ačkoli funguje s mnoha různými programovacími jazyky, je často používán s PHP [61].

2.14 Správa verzí

Správa verzí je systém, který zaznamenává změny souborů v průběhu času a uživatel má možnost obnovit jejich konkrétní verzi. Lze provádět na téměř jakémkoli typu souboru. Umožňuje vrátit jednotlivé soubory do předchozího stavu, porovnávat změny, kdo je autorem jaké verze a mnoho dalšího. Lze tedy snadno získat ztracené soubory nebo vrátit nežádané úpravy [62].

2.14.1 Lokální systémy správy verzí

Lokální systém správy verzí disponují jednoduchou lokální databází, ve které se uchovávají všechny změny souborů s nastavenou správou verzí. Jedním z oblíbených nástrojů tohoto typu byl systém „rcs“. Tento nástroj pracuje tak, že uchovává ve speciálním formátu seznam změn mezi jednotlivými verzemi. Systém později může na základě porovnávání těchto změn vrátit jakýkoliv soubor do podoby, ve kterém byl

v jakékoliv verzi. Toto řešení obsahuje to riziko, že pokud dojde k trvalé ztrátě databáze lokální správy verzí, dojde ke ztrátě celé historie projektu [62].

2.14.2 Centralizované systémy správy verzí

Centralizované systémy správy verzí mají oproti lokálním systémům tu výhodu, že řeší problematiku spolupráce s dalšími pracovníky v týmu. Mezi tyto systémy patří například CVS, Subversion nebo Perforce. Tyto systémy obsahují mimo lokální část i serverovou, která uchovává všechny verzované soubory. Z tohoto centrálního úložiště si potom soubory stahují jednotliví klienti [62].

Oproti lokálním systémům mají mnoho výhod, například, že do určité míry vědí všichni účastníci projektu co kdo na projektu realizuje a administrátoři mají přesnou kontrolu nad jednotlivými právy. Díky tomu je i podstatně jednodušší spravovat centralizované systémy, než pracovat s lokálními databázemi na jednotlivých klientech [62].

Bohužel tento koncept má závažné nedostatky. Mezi největší nedostatek tohoto systému je ten fakt, že pokud vypadne centrální server, nelze během výpadku buď pracovat vůbec nebo přinejmenším není možné ukládat změny ve verzích souborů. Pokud pak dojde ke zničení úložiště, na němž je uložena databáze centralizovaného systému správy verzí a toto úložiště není předem zálohované, dojde ke ztrátě všech dat (celé historie projektu) – podobně jako u lokálního systému správy verzí [62].

2.14.3 Distribuované systémy správy verzí

V případě distribuovaných systémů správy verzí uživatelé ze serveru nestahují pouze nejnovější verzi souborů (snímek), ale uchovávají kompletní kopii celého repositáře. Pokud tedy dojde ke kolapsu serveru (i trvalému bez záloh), lze jej znovu obnovit zkopírováním repositáře od jakéhokoliv uživatele. Každá lokální kopie je pak plnohodnotnou zálohou všech dat. Mnoho systémů tohoto typu navíc dokáže pracovat i s několika vzdálenými repositáři, lze tedy v rámci jednoho projektu spolupracovat

s několika skupinami lidí najednou. Mezi tyto systémy patří například Git, Mercurial, Bazaar nebo Darcs [62].

2.15 Composer

Composer je nástroj pro správu závislostí v PHP. Umožňuje deklarovat knihovny, na kterých váš projekt závisí, a bude je spravovat (instalovat / aktualizovat) [63].

Je nejdůležitějším nástrojem pro PHP vývojáře [64].

„Composer je PHP skript zabalený do souboru composer.phar a pro svůj běh vyžaduje nainstalované PHP minimálně ve verzi 5.3.2. Dalším požadavkem může být přítomnost softwaru git, svn, fossil nebo hg v závislosti na tom, v čem jsou požadované balíčky verzované, ale na to by měl Composer v případě potřeby upozornit“ [65].

Předpoklady pro použití:

- projekt závisí na jiných knihovnách,
- tyto knihovny závisí na dalších knihovnách [63].

Funkce Composeru:

- umožňuje deklarovat knihovny, na které je projekt vázán,
- zjistí, které verze balíčků mohou a musí být nainstalovány a nainstaluje je (což znamená, že je stáhne do projektu) [63].

2.15.1 Satis

Satis je open-source generátor statického Composer repositáře. Je to ultralehké řešení, které umožňuje hostování metadata privátních balíčků (repositářů) [66].

2.16 Nette Framework



Obrázek 2.4: Logo Nette Framework, zdroj: [67]

„Nette je rodina vyspělých a samostatně použitelných komponent pro PHP. Nechte se jimi nadchnout. Dohromady tvoří framework, vyhodnocený jako 3. nejpopulárnější na světě. Filosofie Nette klade mimořádný důraz na produktivitu, nejlepší postupy a bezpečnost“ [68].

Nette Framework je vytvořený tak, aby byl co nejpoužitelnější. Jde o framework, se kterým je snadné a zábavné pracovat. Dává srozumitelnou syntaxi, napomáhá při programování a debugování a pomáhá soustředit se na kreativní stránku vývoje [69].

Nette Framework používají významné společnosti jako třeba T-Systems, GE Money, Mladá fronta, DHL, Logio, ESET, Actum, Slevomat, Socialbakers, SUPRA-PHON a další [69].

2.17 Nittro

Nittro je JavaScriptový framework na straně klienta, speciálně vytvořený pro Nette aplikace. Vytváří kompletní AJAXem řízené stránky. Stará se o základní funkcionality, jako jsou aktualizace sekcí na stránce, aktualizace historie prohlížeče, zobrazování animací během načítání obsahu, validace formuláře na klientské straně, AJAX upload, zobrazování systémových zpráv a další. Lze do Nette aplikace snadno aplikovat a neobsahuje žádné závislosti na další JavaScriptové knihovny [70].

2.18 Doctrine 2



Obrázek 2.5: Logo Doctrine, zdroj: [71]

Doctrine 2 je objektově-relační mapovač (ORM) pro PHP 7.1+, který zajišťuje transparentní ukládání pro objekty PHP. V jádře využívá model datového mapovače, jehož cílem je úplné oddělení domény (obchodní logiky) od logiky ukládání dat do relační databáze [72].

Výhodou Doctrine pro programátora je schopnost soustředit se na objektově orientovanou obchodní logiku a starat se o ukládání dat do relační databáze pouze jako sekundární problém. Ukládání dat do relační databáze je tedy zcela řešeno pomocí Doctrine [72].

2.18.1 Entity

„Entity jsou základní kameny v Doctrine 2 a vůbec celé vaší aplikace. Každá entita reprezentuje nějaký objekt reálného světa, takzvaný doménový objekt. Jednu entitu tak budu mít definovanou pro článek, jinou pro kategorii, další pro uživatele“ [73].

Entity jsou tedy v PHP objekty které lze identifikovat na základě mnoha požadků pomocí jedinečného identifikátoru nebo primárního klíče. Tyto třídy nepotřebují rozšířit žádnou abstraktní základní třídu nebo rozhraní. Třída entity nesmí být konečná nebo nesmí obsahovat konečné metody [72].

2.19 Elasticsearch



Obrázek 2.6: Logo Elasticsearch, zdroj: [74]

Elasticsearch je vysoce škálovatelný open-source fulltextový vyhledávací a analytický engine. Umožňuje rychle a v reálném čase ukládat, vyhledávat a analyzovat velké objemy dat. Používá se obecně jako základní motor / technologie, která napomáhá aplikacím, které mají složité vyhledávací funkce a požadavky [75].

Pár vzorových případů pro využití Elasticsearch:

- díky uložení celého katalogu do Elasticsearch lze produkty vyhledávat a automaticky doplňovat vyhledávání v e-shopu,
- shromažďování a následná analýza a vyhodnocování protokolů (hledání trendů, statistik, agregační funkce, vyhledávání anomálií),
- řešení analytických a Business-intelligence potřeb a je kladen důraz na rychlou analýzu a vizualizaci dat mezi miliony nebo miliardy záznamů [75].

2.20 Message broker

MessageBroker je prostřední počítačový programový modul, který převádí zprávu z formálního protokolu zasílání odesílatele do formálního protokolu zasílání zpráv příjemce. Zprostředkovatelé zpráv jsou prvky v telekomunikačních nebo počítačových sítích, kde softwarové aplikace komunikují výměnou formálně definovaných zpráv [76].

2.20.1 RabbitMQ



Obrázek 2.7: Logo RabbitMQ, zdroj: [77]

RabbitMQ je nejrozšířenější open-source message broker. Běží na mnoha operačních systémech a cloudových prostředích a poskytuje širokou škálu vývojářských nástrojů pro většinu populárních jazyků. Je lehký a snadno se rozšiřuje i v cloudových řešeních. Lze nasadit na distribuované i sdružených konfiguracích tak, aby splňovaly vysoké požadavky na vysokou dostupnost [78].

2.21 Virtualizace

„Virtualizace vytváří simulované neboli virtuální výpočetní prostředí namísto fyzického prostředí. Virtualizace často zahrnuje počítačem vytvořené verze hardwaru, operačních systémů, úložných zařízení atd. To umožňuje organizacím rozdělit jeden fyzický počítač nebo server na několik virtuálních počítačů. Každý virtuální počítač pak může pracovat nezávisle a spouštět různé operační systémy nebo aplikace, zatímco sdílí prostředky jediného hostitelského počítače“ [79].

2.21.1 Vagrant

Vagrant je nástroj pro vytváření a správu prostředí virtuálních počítačů. Poskytuje snadno konfigurovatelné, reprodukovatelné a přenosné pracovní prostředí postavené na špičkové technologii a řízené jediným konzistentním pracovním postupem, který pomáhá maximalizovat produktivitu jednotlivce i celého vývojového týmu [80].

2.21.2 Laravel Homestead

Laravel Homestead je oficiální, předem zabalený Vagrant box, který poskytuje skvělé vývojové prostředí bez nutnosti instalace PHP, webového serveru a jakéhokoliv jiného serverového softwaru na lokálním počítači [81].

Aktuálně Laravel Homestead ve verzi 5.8 obsahuje například software:

- Ubuntu 18.04,
- Git,
- PHP 7.3,
- Nginx,
- MySQL,
- Composer,
- Node (Yarn, Bower, Grunt, a Gulp),
- Redis,
- Elasticsearch,
- a další [81].

2.22 Analýza statického kódu

Analýza statického kódu (známá také jako analýza zdrojového kódu) se obvykle provádí jako součást kontroly kódu (také označovaného jako testování bílého boxu). Analýza statického kódu se běžně vztahuje k nástrojům pro analýzy, které se pokoušejí rozpoznat možné chyby v rámci testování „statického“ (nespuštěného) zdrojového kódu pomocí technik, jako je například analýza barev a analýza toku dat [82].

2.22.1 Cyklomatická složitost

Cyklomatická složitost je metrika kvality softwaru. Nejjednodušší definice popisuje cyklomatickou složitost jako počet rozhodnutí ve zdrojovém kódu. Čím vyšší je počet, tím složitější je kód. Je důležitou metrikou, protože určuje limitní komplexnost kódu a určuje počet požadovaných testovacích případů [83].

Kapitola 3

Analýza současné situace

V této části práce se budu věnovat popisu základních informací o živnostníkovi a popisu aktuálního stavu, ve kterém se jeho e-shopové řešení nachází.

3.1 Základní informace o živnostníkovi

„Hana Hladíková (68737424) je Fyzická osoba podnikající dle živnostenského zákona nezapsaná v obchodním rejstříku. Má celkem jednu živnost. Živnost provozuje od roku 2001. V současnosti stále podniká“ [84].

Souhrnné informace:

- jméno: Hana Hladíková,
- právní forma podnikání: fyzická osoba podnikající dle živnostenského zákona nezapsaná v obchodním rejstříku,
- přidělený úřad: městský úřad Velké Meziříčí,
- vznik živnosti: 1.7.2001,
- sídlo: Lavičky 127, 594 01,
- klasifikace ekonomických činností -CZ-: ostatní profesní, vědecké a technické činnosti, poradenství v oblasti řízení,

- IČO: 68737424,
- DIČ: CZ7656024827,
- datum registrace k dani z přidané hodnoty: 13.3.2019,
- webové stránky: HanaBrand.cz.

3.2 Značka HanaBrand

V rámci živnostnickovy podnikatelské činnosti vznikla v roce 2015 značka HanaBrand a pod touto značkou jsou označovány všechny nabízené produkty.



Obrázek 3.1: Logo HanaBrand používané převážně na výrobcích, zdroj: živnostnickovy interní materiály

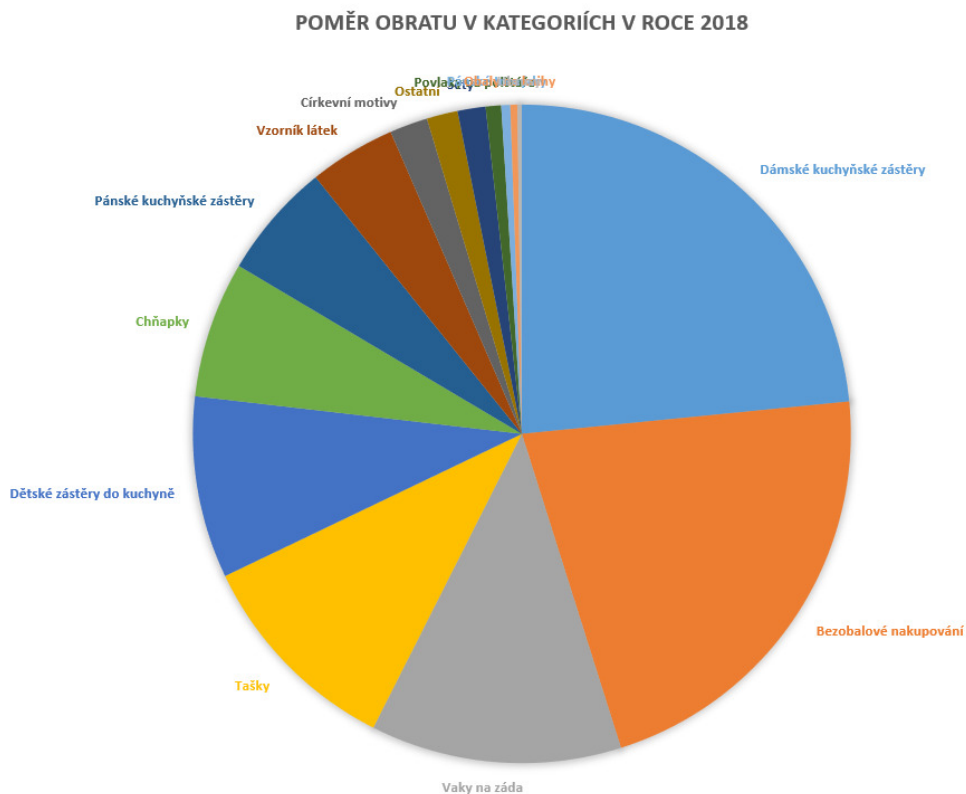


Obrázek 3.2: Zjednodušená varianta loga HanaBrand, zdroj: živnostnickovy interní materiály

3.3 Nabízené produkty a služby

Živnostník pod značkou HanaBrand nabízí produkty vlastní výroby, které jsou vyrobeny z certifikovaných materiálů českého původu. Mezi tyto produkty patří například:

- dámské kuchyňské zástěry,
- ekologické pytlíky (pro bezobalové nakupování),
- vaky na záda,
- tašky,
- dětské zástěry do kuchyně,
- chňapky,
- pánské kuchyňské zástěry.



Obrázek 3.3: Poměr obrátu v jednotlivých kategoriích na e-shopu v roce 2018, zdroj: interní živnostníkovy dokumenty

Mimo produkty také živnostník nabízí službu zakázkového šití.

3.4 Obchodní vztahy

Živnostník nabízí pod značkou HanaBrand své vlastní produkty a služby formou B2C a B2B.

3.4.1 B2C

Výrobky a služby živnostník poskytuje koncovým zákazníkům prostřednictvím následujících kanálů:

- vlastní E-shop,
- handmade internetové tržiště Fler.cz,
- handmade internetové tržiště Sashe.sk,
- přímý prodej (trhy, jarmarky, zakázková výroba).

3.4.2 B2B

Výrobky a služby živnostník poskytuje obchodním partnerům pouze prostřednictvím e-mailové a telefonické komunikace. Vytvoření objednávky s velkoobchodní slevou prostřednictvím e-shopu není možné, protože aktuální e-shopové řešení touto funkcionalitou nedisponuje. Zároveň poskytuje společností na Vysočině komisioní prodej.

3.5 Současná technická situace

V této části bude popsána současná technická situace ve kterém se nachází aktuální e-shopové řešení. Aktuálně provozované e-shopové řešení bylo vytvořeno v roce 2016. Jedná se o PHP aplikaci, která je objektově orientovaná a využívá návrhový vzor MVC. Řešení nevyužívá žádný aplikační framework, proto je poměrně složité toto řešení snadno rozšiřovat tak, aby splňoval veškeré požadavky živnostníka.

3.5.1 Technické požadavky současného e-shopu

Pro provoz současného e-shopového řešení jsou potřeba následující technické požadavky:

- Apache Server 2.4.29,
- PHP 5.6.40,
- rozšíření webového serveru mysqli, gd a json,
- MariaDB 10.0.21.

3.5.2 Entity v aplikaci

Seznam existujících entit v aplikaci:

- Article (článek),
- ArticleCategory (kategorie článků),
- Basket (košík),
- Category (kategorie produktů),
- CustomField (uživatelské pole produktu),
- CustomFieldValue (hodnota uživatelského pole produktu),
- Discount (sleva),
- Order (objednávka),
- OrderItem (produkt objednávky),
- Payment (platba),
- Product (produkt),
- Shipping (doprava),
- User (uživatel).

3.5.3 Administrované entity v aplikaci

Bjeme kuchyňské zástěry a doplňky pro zdešení domova. Z kvalitních látek a s láskou. +420 602 763 262 PODPOR@HABBA.BRAND.CZ FACEBOOK JAK NAKUPOVAT KONTAKT

HABBA ÚVOD VÝROBKÝ O NÁS SPOLUPRÁCE KONTAKT Kč 0,-

PŘIHLÁŠENÍ UŽIVATELE

Přihlašovací jméno

Přihlašovací jméno

PŘIHLÁSIT

HLAVNÍ NABÍDKA
VÝROBKÝ
SPOLUPRÁCE
KONTAKT
JAK NAKUPOVAT

PRODUKTY
ZÁSTĚRY PRO ŽENY
ZÁSTĚRY PRO DĚTI
ZÁSTĚRY PRO MUŽE
PÁNSKÉ TRENGYRY
BEZBOALOVÉ NAKUPOVÁNÍ
VÁKY NA ŽÁDA

CIRKEVNÍ MOTIVY
SETY
POVLAKY NA POLSTARE
OHLASKY
UBRUSY

OBALY NA KNHÝ
TÁSKY
OSTATNÍ
VSECHNY PRODUKTY

LÁTKY
VÁŠ KOSK
JAK NAKUPOVAT
PŘIDÁVNKY NÁKUPU
INFORMACE O DOPRAVĚ

Copyright © 2019 - Petr Hradík
+420 602 763 262 | podpora@habba-brand.cz | [jak nakupovat](#) | [Přihlas na Facebooku](#)

Obrázek 3.4: Přihlášení do administračního rozhraní e-shopu, zdroj: současné e-shopové řešení

Seznam entit, které jsou administrovatelné v administrační části e-shopu:

- Discount (sleva) – pouze vytvoření,
- Order (objednávka),
- Product (produkt).

V administraci produktů lze konfigurovat vlastnosti, které definují obsah produktu. Lze nastavit:

- název,
- alias (část URL adresy pro zobrazení detailu produktu),
- zařazení do kategorie,
- stav produktu (nezveřejněný, zveřejněný, vyprodaný),
- příznak doporučeného produktu,
- příznak nového produktu,
- počet návštěvníků, kterým se produkt líbil (ikona srdíčka na detailu produktu),
- konfigurace uživatelských polí (popis, fotografie, fyzické proporce, barva, vzor, atd.),
- metadescription,
- zobrazení v exportních feedech,
- název v exportních feedech (pokud se liší od základního názvu).

<

Obrázek 3.7: Přehled objednávek v administraci, zdroj: současné e-shopové řešení

Správa objednávky je o poznání prostější. Objednávce lze v administračním rozhraní nastavit pouze stav. Na výběr jsou možnosti:

- nová,
- zpracovává se,
- zaplaceno,
- odesláno,
- dokončeno,
- stornováno.

Při změně objednávky na stavy „zaplaceno“, „odesláno“, „dokončeno“ odešle systém zákazníkovi e-mail zákazníkovi s informacemi o současném stavu jeho objednávky.

[Přehled produktů](#)
[Přehled objednávek](#)

[Jste přihlášen/a jako super adminstrátor](#)
[Odhlásit](#)

Býme kuchyňské zástěry a doplňky pro zkrášlení domova. Z kvalitních látek s láskou.

+420 605 763 262
 podpora@hanabrand.cz
[FACEBOOK](#)
[JAK NAKUPOVAT](#)
[KONTAKT](#)

[ÚVOD](#)
[VÝROBKÝ](#)
[O NÁS](#)
[SPOLUPRÁCE](#)
[KONTAKT](#)

0,-

Stav objednávky

Nová

Stavy s vyklíčením odešlou zákazníkovi informaci o změně stavu e-mailem.

ULOŽIT STAV

SOUHRNĚ INFORMACE

ID objednávky	20190043
datum objednání	2019-03-16 13:33:55
stav objednávky	Nová
Celková cena	369 Kč
Sleva	0 Kč
Způsob platby	Platba převodem
Variabilní symbol	120190043
Způsob Dopravy	Česká pošta - doporučený dopis
Jméno	
Příjmení	
E-mail	
Ulice	
Město	
PSČ	
Telefon	
Poznámka	
IČO	
DIČ	

OBJEDNANÉ ZBOŽÍ

	Jméno	Počet	Celková cena za položku
1	Vak na zedra - Kočky	1	299 Kč

INFORMACE O SLEVĚ

Nebyla aplikována žádná sleva

INFORMACE K OBJEDNÁVCE

- žádné informace -

HLAVNÍ NABÍDKA

[VÝROBKÝ](#)
[SPOLUPRÁCE](#)
[KONTAKT](#)
[JAK NAKUPOVAT](#)

PRODUKTY

[ZÁSTĚRY PRO ŽENY](#)
[ZÁSTĚRY PRO DĚTI](#)
[ZÁSTĚRY PRO MUŽE](#)
[PÁNSKÉ TRENOVÁKY](#)
[SENZIGALOVÉ NAKUPOVÁNÍ](#)
[VÁKY NA ŽÁDA](#)

OKRSEVNÍ MOTIVY

[SETY](#)
[POVLAKY NA POLSTARE](#)
[CHRÁPKY](#)
[UBRUSY](#)

OBALY NA KNHÝ

[TASKY](#)
[OSTATNÍ](#)
[VŠECHNY PRODUKTY](#)

LÁTKY

[VÁL KOSK](#)
[JAK NAKUPOVAT](#)
[PODMINKY NAKUPU](#)
[INFORMACE O DOPRAVĚ](#)

Copyright © 2019 - Petr Hladík
 +420 605 763 262
 podpora@hanabrand.cz
[jak nakupovat](#)
[Přidat na Facebooku](#)

Obrázek 3.8: Detail objednávky, zdroj: současné e-shopové řešení

Formulář pro vytvoření nového slevového kupónu je také velice prostý. Slevové kupóny nelze z pohledu administračního rozhraní systému nijak pokročile spravovat, lze je pouze vytvořit.

Vytvoření nového slevového kupónu

Částka pevná	<input type="text" value="0"/>	↕	Kč	
Částka procentuální	<input type="text" value="0"/>	↕	%	
Minimální částka objednávky	<input type="text" value="0"/>	↕	Kč	
Platnost do	<input type="text" value="30 . 03 . 2019"/>	⊗	<input type="text" value="23 : 59"/>	⊗

Vytvořit

Obrázek 3.9: Formulář pro přidání nového slevového kupónu, zdroj: současné e-shopové řešení

Po definování slevy (pevné částky nebo procentuální sazby z celkové ceny objednávky), minimální celkové částky objednávky pro umožnění aplikace slevy a maximální platnosti kupónu systém vygeneruje náhodný jedinečný slevový kód kupónu a uloží kupón do databáze.

3.5.4 Implementované funkcionality

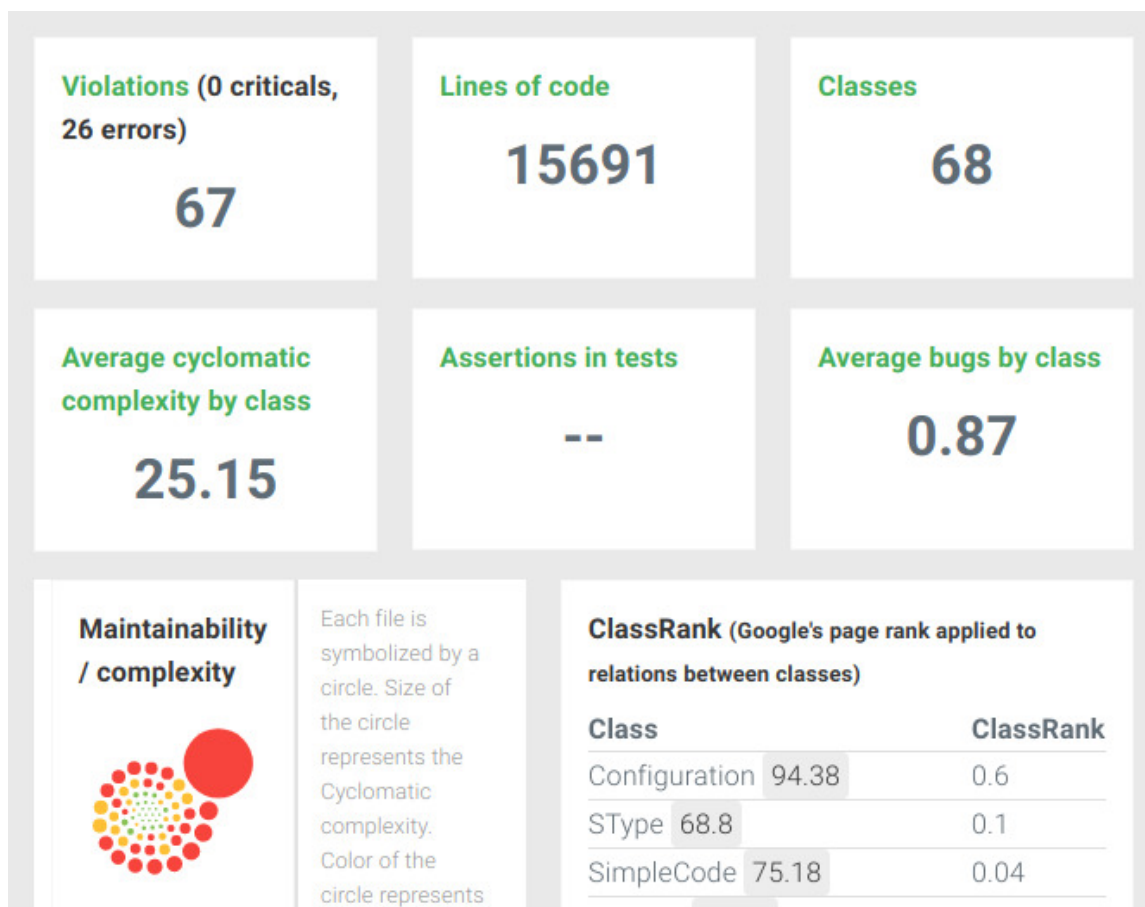
Seznam implementovaných funkcionalit, kterými aktuální řešení disponuje:

- administrace objednávek,
- administrace produktů,
- aplikace slevových kupónů na objednávku,
- export produktů do externích systémů (Zboží.cz, Google Merchants),
- individuální definice cen způsobů doprav,
- individuální definice cen způsobů plateb v závislosti na zvoleném způsobu dopravy,
- odesílání e-mailů administrátorům a zákazníkovi při vytvoření objednávky a při změně stavu objednávky,

- odesílání zprávy přes nástroj Pushover administrátorům při vytvoření nové objednávky,
- responzivní design šablon,
- správa košíku produktů,
- vytvoření a dokončení objednávky,
- vytvoření slevového kupónu,
- výběr parcelshopu v případě zvolení způsobu dopravy přes Zásilkovnu,
- zobrazení detailu produktu,
- zobrazení detailu článku,
- zobrazení kategorie produktů.

3.6 Statická analýza kódu

Protože mám přístup k zdrojovým souborům současně provozovaného řešení, mohu provést statickou analýzu kódu za pomoci nástroje PhpMetrics.



Obrázek 3.10: Výsledek statické analýzy kódu současného řešení, zdroj: nástroj PhpMetrics

Z výsledků analýzy lze vyčíst například základní informace, tedy, že zdrojový kód obsahuje:

- 68 tříd,
- 15 691 řádků kódu,
- v tomto kódu se vyskytuje celkem 67 chyb,
- průměrný počet 0,87 chyb na jednu třídu,
- index průměrné cyklomatické složitosti je 25,15.

3.7 Zhodnocení současného stavu

Na základě konzultace s paní Hanou Hladíkovou, provozovatelkou současného e-shopového řešení, lze konstatovat, že zhruba 60 % příjmů tvoří objednávky vytvořené přes současné e-shopové řešení. Živnostník je tedy z větší části závislý na internetovém prodeji skrz současný e-shop.

Toto řešení vzniklo jako jednoduchá aplikace na zelené louce bez využití jakéhokoliv aplikačního frameworku v roce 2016. Od té doby se řešení prakticky nevyvíjelo, funkcionality nevylepšovaly a kód se neaktualizoval, čemuž také nasvědčuje fakt, že nebyla aktualizována verze PHP, na kterém je systém provozován (aplikace pro svůj chod používá PHP 5.6.40). Zároveň zdrojové kódy e-shopu neobsahují například dnes již běžně používané jmenné prostory, které slouží ke sdružování logických celků tříd aplikace.

Sám živnostník cítí potřebu inovace řešení, protože současné řešení postrádá spoustu důležitých funkcí. Inovací řešení očekává především usnadnění práce s e-shopem, přehlednější a snadnější administrativu. E-shop by měl být pro zákazníky sestaven tak, aby se v něm dobře orientovali a například jim umožnit vkládat recenze k produktům, čímž se také zvýší důvěryhodnost internetového obchodu. Dále plánuje nabízet své výrobky a služby za hranicemi České republiky, z toho důvodu požaduje lokalizaci celého řešení.

V současnosti živnostník propaguje svoje výrobky pomocí placených kampaní Sklik, Zboží.cz, Facebook i Instagram. Po inovaci řešení má v plánu investovat do marketingu mnohem více finančních prostředků, než kolik investuje nyní, což se má projevit na návštěvnosti e-shopu a zvýšení konverzního poměru.

Celé řešení by tak mělo usnadňovat živnostníkovi práci, umožnit konfigurovat systém dle potřeb, poskytovat potřebné funkcionality, které v současném řešení chybí a napomocť s některými činnostmi jejich automatizací.

3.8 Požadavky na nové řešení

Na základě konzultace s Hanou Hladíkovou, byl sestaven následující seznam požadavků, které by mělo nové řešení poskytovat:

- administrace všech entit v systému (nejen produkty a objednávky),
- propojení s cenovými srovnávači (Zboží.cz, Heuréka, Google Merchants),
- platba kartou,
- varianty produktů,
- uživatelské parametry produktů,
- propracovaná cenotvorba (měny, slevy, cenové hladiny),
- konfigurace dostupných způsobů doprav (dle zemí, dle ceny objednávky, dle rozměrů a váhy košíku),
- zobrazování cen s DPH i bez DPH,
- propojení s externími službami (Zásilkovna, Česká pošta),
- správa skladů,
- dodací i fakturační adresa pro objednání,
- uživatelské účty pro zákazníky (rezervace produktů, historie objednávek, přiřazení cenové hladiny, správa adres),
- recenze produktů,
- správa obsahu (bannery, články, kategorie),
- překlady do cizích jazyků,
- změna vzhledu e-shopu,
- správa metatagů pro SEO,
- vyhledávání (fulltextové, na základě parametrů).

3.9 Průzkum dostupných řešení

Dané požadavky živnostníka lze splnit pomocí různých dostupných řešení. Ty lze rozdělit dle několika kritérií na:

- vlastní řešení,
- open-source řešení,
- řešení na míru,
- pronájem [85].

3.9.1 Vlastní řešení

Výhodou vlastního řešení je ta, že všechny funkce systému budou pod kontrolou. Když bude potřeba něco změnit, nemusí se na nikoho čekat nebo někomu platit, prostě lze danou funkci do systému doplnit. Dané řešení navíc bude vyřešené na míru konkrétním potřebám dané společnosti [85].

Nevýhodou toho řešení je ale vysoký náklad v podobě velkého počtu stráveného časem při vývoji. Pokud je vývojář daného řešení zároveň majitelem obchodu, doporučuje se spíše investovat strávený čas do samotného rozvoje podnikání, než tvorbou platformy. Pro vývoj vlastní platformy je také vhodné mít již zkušenosti s vývojem e-commerce řešení [85].

Výhody:

- nulové finanční náklady,
- řešení přesně podle potřeb,
- žádná vazba na poskytovatele [85].

Nevýhody:

- velké časové náklady,

- nutnost know-how,
- uvázání se k vývoji [85].

3.9.2 Open-source řešení

Poměrně často se lze na internetu setkat s open-source e-shopovými řešeními. Jejich zdrojové soubory jsou volně ke stažení a systém lze po instalaci na hostingu zprovoznit. Spuštění systému je tedy poměrně velice rychlé. Většinou pro tato řešení zároveň existují různé doplňky od jiných vývojářů (například platba kartou), je tedy možné systém s jejich pomocí customizovat a dále rozšiřovat [85].

Problémem tohoto řešení je, že systém musí být univerzálním řešením pro tisíce obchodníků, tudíž v zásadě nebude systém v základním nastavení vyhovovat individuálním potřebám podniku a bude nutné ho tedy naprosto vždy upravit a doplnit chybějící funkce. Bohužel zároveň většina open-source řešení tohoto typu má velmi špatnou kvalitu zdrojového kódu a mnohdy je i zázrakem, že dané řešení nějakým způsobem vůbec funguje. Některá řešení jsou navíc neuvěřitelně náročná na výkon serveru. Takováto řešení navíc často nedokážou naplnit očekávání dle českých specifik (například práci s dvojí hladinou DPH nebo vystavení faktury dle našich zvyklostí) [85].

Výhody:

- zdarma,
- okamžitě k použití,
- někdy velké komunity pro vývoj dalších funkcí [85].

Nevýhody:

- často dost špatná kvalita kódu,
- náročnost na hosting,

- tuctový vzhled obchodu,
- často nevhodné pro české podmínky,
- zbytečné funkce [85].

Mezi tato řešení patří například:

- WooCommerce,
- X-Cart,
- Zen Cart,
- Magento Open Source,
- OpenCart,
- PrestaShop,
- a další [86].

3.9.3 Řešení na míru

Řešení na míru vznikne tak, že externí společnost vytvoří e-shopové řešení od základů přesně podle požadavků a představ zadavatele. Většinou se vyplatí až pro velké obchody, které mají speciální požadavky a do budoucna je potřeba počítat s náklady na servis [85].

Výhody:

- odladěné funkce,
- vytvoření přesně podle požadavků [85].

Nevýhody:

- velmi vysoká cena [85].

3.9.4 Pronájem

Princip pronajímaného řešení je ten, že dodavatel poskytuje řešení e-shopu obchodníkovi a ten mu za jeho využívání platí měsíční paušál [85].

Mezi výhody tohoto řešení patří cena. Ta se dnes pohybuje v rozmezí stovek korun za měsíční pronájem řešení se základní funkcionalitou. I u tohoto řešení platí, že čím náročnější potřeby a požadavky na řešení, tím vyšší cena za pronájem. Mezi výhody dále patří rychlost spuštění obchodu a pokud poskytovatel tyto služby nabízí, tak i zákaznická podpora a průběžný vývoj [87].

Nevýhodami tohoto řešení jsou limity platformy a platí tedy, že mnohdy se obchodník musí přizpůsobovat e-shopu a ne e-shop obchodníkovi. Při dokupování nadstandardních modulů a úpravách na míru navíc roste i cena řešení (jeho hlavní výhoda). Další nevýhodou je menší možnost úprav a jistá šablonovanost e-shopu. Díky tomu není toto řešení vhodné pro střední a velké e-shopy, které vyžadují spíše robustnější e-shopový systém [87].

Výhody:

- cena,
- rychlost spuštění obchodu [87].

Nevýhody:

- limity platformy,
- přizpůsobení se e-shopu,
- dokupování nadstandardních modulů,
- nákladné úpravy na míru,
- malá možnost úprav,
- šablonovanost systému [87].

3.9.5 Výsledek průzkumu

Živnostník má za sebou již od roku 2015 zkušenosti s různými e-shopovými řešeními, se kterými postupem času nebyl příliš spokojený. Tato řešení se neinovovaly a postupně přestávaly dostačovat potřebám živnostníka. Kombinace těchto faktů s vyhotoveným průzkumem dostupných řešení lze vyvodit závěr, že v aktuální situaci by bylo vhodné, aby bylo pro živnostníka vyhotoveno řešení na míru přesně dle jeho požadavků.

Kapitola 4

Návrh řešení

V této kapitole se zaměřím na návrhovou a implementační část (na úrovni prototypu) vytvoření e-commerce platformy. Tato platforma bude vytvořena tak, aby poskytovala kvalitní technologický základ pro budoucí vývoj v plnohodnotný produkt, který živnostníkovi poskytne náhradu za aktuální e-shopové řešení a bude disponovat funkcionalitami, které od nového řešení požaduje.

Z pohledu technologického zpracování lze vytvořit technicky univerzální řešení, které bude možné customizovat dle přesných konkrétních požadavků. Toto řešení následně živnostníkovi poskytnout formou pronájmu nebo prodejem licence. Pokud bude celé řešení takto koncipováno, bude možné výsledné necustomizované řešení poskytnout i dalším subjektům.

4.1 Výběr technologií

Pokusím se blíže přiblížit vybrané technologie, které tvoří základ připravovaného e-commerce řešení. Veškeré použité technologie jsou poskytovány buď jako open-source nebo zcela zdarma, například pod licencí Apache.

4.1.1 Technologická platforma

Na základě osobních zkušeností jsem se rozhodl realizovat celý projekt za použití programovacích jazyků PHP a JavaScript. PHP pro využití ve skriptování na straně

serveru, JavaScript pro skriptování na straně klienta – pro zvýšení uživatelského požitku z využívání aplikace, například za použití technologie AJAX. Typ databáze jsem zvolil MySQL a to proto, že ho přímo obsahuje připravené vývojové prostředí Laravel Homestead, podporuje ho ORM Doctrine 2 a díky kladným osobním zkušenostem.

Celý projekt jsem se rozhodl pojmenovat jako „Jarvis“.



Obrázek 4.1: Logo systému Jarvis, zdroj: vlastní zpracování

4.1.2 Laravel Homestead

Pro vývojové prostředí byl použit Laravel Homestead, které poskytuje předinstalovaný systém se všemi základními balíčky potřebnými pro vývoj PHP aplikace. To umožňuje poměrně rychlý způsob, jak začít vyvíjet aplikaci bez nutnosti zdlouhavé konfigurace prostředí. Navíc toto řešení disponuje tou výhodou, že v případě vícečlenného týmu vývojářů budou všichni vývojáři disponovat stejným vývojovým prostředím. Bylo tedy možné se okamžitě věnovat realizaci projektu a nebylo nutné trávit čas zbytečnou konfigurací vývojového prostředí.

4.1.3 Composer

Pro co nejvíce systémové řešení jsem se rozhodl celou platformu rozdělit na jednotlivé balíčky (logické celky), které jsou verzovány v samostatných privátních repositářích. Jejich integraci do celistvé aplikace řeší nástroj Composer, jako instalaci jednotlivých závislostí. Každý balíček pak vyžaduje vlastní další závislosti (například na balíčcích třetích stran), které Composer při instalaci systému sám nainstaluje. Navíc Composer při instalaci vytvoří mapu tříd, které se nacházejí v jednotlivých balíčcích a zajistí

jejich automatické načítání během chodu aplikace. To práci výrazně ulehčí, protože není nutné řešit import tříd v aplikaci samotné.

4.1.4 Satis

Aby mohl Composer instalovat balíčky z privátních repositářů, je zapotřebí pro tyto balíčky vytvořit statický Composer repositář. K tomu slouží open-source nástroj Satis. V něm se jednoduše definují zdroje (například GIT repositář), ze kterých Satis vytvoří svůj vlastní statický repositář.

```
{
  "name": "Composer repository",
  "homepage": "http://satis.local",
  "output-dir": "web",
  "output-html": true,
  "repositories": [
    {
      "type": "git",
      "url": "https://gitlab.com/digilabscz/jarvis-shop.git"
    },
    {
      "type": "git",
      "url": "https://gitlab.com/digilabscz/jarvis-core"
    },
    {
      "type": "git",
      "url": "https://gitlab.com/digilabscz/jarvis-cms.git"
    },
    {
      "type": "git",
      "url": "https://gitlab.com/digilabscz/jarvis-app.git"
    },
    {
      "type": "git",
      "url": "https://gitlab.com/digilabscz/doctrine-form-mapper.git"
    }
  ],
  "require-all": false,
  "require-dependencies": false,
  "require-dev-dependencies": false,
  "minimum-stability": "dev",
  "providers": false
}
```

Obrázek 4.2: Satis konfigurace, zdroj: vlastní zpracování

4.1.5 Nette Framework

Na základě zkušeností z předchozího řešení je nutné, aby celé řešení bylo postavené na kvalitním PHP Frameworku, který bude umožňovat poměrně snadný vývoj celé aplikace a do budoucna dostatečný prostor pro snadný rozvoj aplikace. V tomto řešení jsem na základě osobních zkušeností zvolil český Nette Framework v aktuálně poslední stabilní verzi 2.4.

4.1.6 Nittro

Pro lepší uživatelský požitek z užívání aplikace je využita technologie AJAX. Nittro je rozšíření Nette Framework, které automaticky umožní v aplikaci využít asynchronní zpracovávání HTTP požadavků. Toho je využito hlavně v administrační části systému, kde jsou naprosto všechny požadavky asynchronní. To zrychluje práci se systémem, protože není nutné čekat na překreslení celého dokumentu webové stránky.

4.1.7 Doctrine 2

Objektový přístup k entitám modelové vrstvy systému zajišťuje objektově relační mapování. Toho je v systému dosaženo použitím Doctrine 2, který mapuje entity systému na data uložená v MySQL databázi. Také zodpovídá za strukturu databáze. Informace o schématu relace v databázi čte přímo v třídách entit, kde jsou tyto informace zapsány pomocí speciálního anotačního zápisu.

```
<?php

namespace DigilabsCz\JarvisCore\Models\Entities;

use DigilabsCz\JarvisCore\Doctrine\Types\NetteDateTimeType;
use Doctrine\ORM\Mapping as ORM;
use Kdyby\Doctrine\Entities\Attributes\Identifier;
use Nette\Utils\DateTime;

/**
 * @ORM\Table(name="address", indexes={@ORM\Index(name="country", columns={"country"})})
 * @ORM\Entity
 * @ORM\HasLifecycleCallbacks
 * @ORM\Entity(repositoryClass="DigilabsCz\JarvisCore\Models\Repositories\AddressRepository")
 */
class AddressEntity extends BaseEntity
{
    use Identifier;

    /**
     * @var string
     *
     * @ORM\Column(name="email", type="string", length=64, nullable=true)
     */
    protected $email;
```

Obrázek 4.3: Ukázka zdrojového kódu s anotačním zápisem pro Doctrine 2, zdroj: gitlab.com

4.1.8 Elasticsearch

Pro vyhledávání v systém jsem využil fulltextový vyhledávací engine Elasticsearch. Ten zajišťuje například vyhledávání produktů nebo článků v modulu systému, který zobrazuje data zákazníkům obchodu. Vyhledávání je možné provést nad všemi daty,

které jsou ze systému vyexportovány a naimportovány do Elasticsearch. Je tedy například možné vyhledávat pomocí titulku nebo konkrétního parametru entity.

4.1.9 RabbitMQ

RabbitMQ je používán jako fronta pro asynchronní paralelní zpracovávání úloh v systému. Řeší například odesílání e-mailů nebo automatický export entit do Elasticsearch. Výhodou tohoto řešení je fakt, že konkrétní úlohy se nemusí zpracovávat ve stejném vlákne a dojde ke zrychlení odpovědi systému na konkrétní požadavek, protože úlohy jsou zpracovány paralelně v samostatných vláknech. Zároveň je možnost větší kontroly nad chybami během zpracovávání úloh. Pokud například dojde k výskytu chyby během zpracovávání úlohy, daná úloha není ve frontě označena jako zpracovaná, ale po určitém časovém intervalu se RabbitMQ opět pokusí danou úlohu zpracovat.





4.2 Struktura aplikace

V této sekci blíže popíší strukturu aplikace. Jednotlivé rozdělení do logických celků (balíčků) a přiblížení adresářové struktury.

4.2.1 Části systému

Celý systém je rozdělen na samostatné logické celky, které jsou reprezentovány jako samostatně instalovatelné balíčky a jsou verzované v samostatných repositářích. Tyto balíčky jsou:

- Jarvis App,
- Jarvis Core,
- Jarvis CMS,
- Jarvis Shop.

📁  Jarvis App	★ 0 🗄 11 minutes ago
📁  Jarvis Shop	★ 0 🗄 11 minutes ago
📁  Jarvis CMS	★ 0 🗄 11 minutes ago
📁  Jarvis Core	★ 0 🗄 10 minutes ago

Obrázek 4.4: Rozdělení systému na balíčky – jednotlivé repositáře, zdroj: gitlab.com

Systém je tvořen samostatnými balíčky, které se instalují pomocí nástroje Composer. Zdrojové kódy těchto balíčků se po instalaci nacházejí v adresáři „vendor/“ a z pohledu aplikace by nemělo docházet k úpravám jejich kódu (úpravy se s aktualizací balíčků zahodí).

Jarvis App

Jarvis App je základním balíčkem aplikace. Balíček je typu „projekt“. Vytváří základní adresářovou strukturu aplikace a je závislý na balíčku Jarvis Core. Pomocí tohoto balíčku lze tedy vytvořit samotnou aplikaci (instalací nadstavby Jarvis CMS nebo Jarvis Shop).



Obrázek 4.5: Logo Jarvis App, zdroj: vlastní zpracování

Jarvis Core

Jádro aplikace, balíček je typu „knihovna“. Vytváří všechny základní prostředky, kterými systém disponuje a zbytek aplikace je na těchto prostředcích závislý. Lze v něm

tedy například nalézt základní modelovou strukturu aplikace, a základní komponenty systému.



Obrázek 4.6: Logo Jarvis Core, zdroj: vlastní zpracování

Jarvis CMS

Balíček typu „knihovna“, poskytuje funkcionality systému pro správu obsahu. Balíček je závislý na balíčku Jarvis Core. Obsahuje modelovou strukturu aplikace a funkcionality, které rozšiřují základní systém o správu obsahu – možnost správy kategorií a článků.



Obrázek 4.7: Logo Jarvis CMS, zdroj: vlastní zpracování

Jarvis Shop

Balíček typu „knihovna“, poskytuje funkcionality e-commerce systému. Balíček je závislý na balíčku Jarvis Core a Jarvis CMS. Obsahuje modelovou strukturu aplikace a funkcionality, které rozšiřují základní systém o funkce elektronického obchodu.

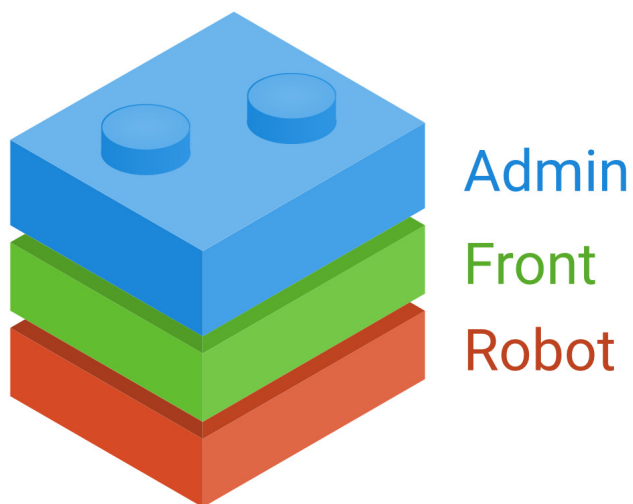


Obrázek 4.8: Logo Jarvis Shop, zdroj: vlastní zpracování

4.2.2 Moduly

Základní systém je ve své vnitřní logice rozdělen na 3 moduly. Ty se vzájemně liší svými účely použití. Tyto moduly jsou:

- Admin,
- Front,
- Robot.



Obrázek 4.9: 3 Moduly systému, zdroj: vlastní zpracování

Admin

Modul Admin poskytuje administrační rozhraní celého systému. Na základě ACL přihlášenému uživateli správu administrovatelných entit, které se v celém systému nacházejí.

Front

Modul Front poskytuje rozhraní, které je výsledkem konfigurace systému v administračním rozhraní. Právě tento modul je zodpovědný za prezentování dat zákazníkům obchodu, tedy mimo jiné například za prezentování zobrazení produktové kategorie návštěvníkům e-shopu, vložení produktu do košíku, vytvoření a dokončení objednávky, zobrazení detailu článku, atd.

Robot

Tento modul poskytuje služby pouze systému samotnému. Například se v tomto modulu nachází logika pro poskytování funkcionality ke asynchronní zpracování obrázků.

4.2.3 Adresářová struktura

Name
app
log
temp
vendor
www
.gitignore
composer.json
composer.lock
init.php
jarvis

Obrázek 4.10: Základní adresářová struktura aplikace, zdroj: náhled repositáře v administraci gitlab.com

Zprovozněná aplikace má následující kořenovou adresářovou a souborovou strukturu:

- app/ – konfigurační soubory, customizace systému (pomocí přepisování tříd a šablon),
- log/ – ukládání logů,
- temp/ – dočasné soubory (cache, doctrine proxy entity),
- vendor/ – nainstalované balíčky (závislosti) pomocí nástroje Composer,
- www/ – veřejně dostupné soubory (média, soubory šablon),
- .gitignore – konfigurace ignorovaných souborů GIT,
- composer.json – konfigurace závislostí aplikace,
- composer.lock – zámek verzí závislostí,

- `init.php` – inicializace aplikace,
- `jarvis` – přístupový bod konzolových příkazů,
- `readme.md` – informace o aplikaci.

4.3 Modelová vrstva

Modelová vrstva aplikace obsahuje samotné entity, repositáře a kontejnery. Tato vrstva je obsažena v jádře systému a využívají ji všechny moduly společně.

4.3.1 Entity

V aplikaci se využívá objektově relační mapování za pomoci Doctrine 2 a tudíž je nutné každou relaci v databázi definovat jako entitu v podobě třídy. Každá entita definuje modelové schéma pomocí anotačního zápisu. Tento anotační zápis je za pomoci Doctrine 2 následně transformován na strukturu relace databáze. Entity v systému tedy definují strukturu relací i datové typy a entita (objekt) je pak nosičem stejných informací, jako je záznam v databázi. Při ukládání stavu entit jsou změny v entitách (objektech) uloženy do databáze.

Seznam entit, které obsahuje balíček Core:

- `Address` – adresa,
- `Country` – země,
- `Image` – obrázek,
- `Language` – jazyk,
- `Menu` – menu,
- `MenuPage` – stránka menu,
- `MessageTemplate` – šablona zpráv,

- Metadata – meta tagy,
- Route – stránka,
- User – uživatel,
- UserGroup – uživatelská skupina,
- UserRole – uživatelská role,
- UserRoleResource – oprávnění uživatelské role.

Seznam entit, které obsahuje balíček CMS:

- Article – článek,
- ArticleCategory – kategorie článků,
- ArticleImage – obrázek článku.

Seznam entit, které obsahuje balíček Shop:

- Currency – měna,
- Coupon – slevový kupón,
- Manufacturer – výrobce,
- Order – objednávka,
- OrderProduct – produkt objednávky,
- OrderProductAttribute – atribut produktu objednávky,
- OrderState – stav objednávky,
- Payment – platba,
- PaymentMethod – způsob platby,
- PaymentMethodType – typ způsobu platby,

- PaymentState – stav platby,
- PriceLevel – cenová úroveň,
- Product – produkt,
- ProductCategory – kategorie produktu,
- ProductComment – komentář k produktu,
- ProductImage – obrázek produktu,
- ProductParameter – parametr produktu,
- ProductPrice – cena produktu,
- ProductRate – hodnocení produktu,
- ProductRateImage – obrázek hodnocení produktu,
- ProductState – stav produktu,
- Shipping – doprava,
- ShippingMethod – způsob dopravy,
- ShippingMethodType – typ způsobu dopravy,
- ShippingState – stav dopravy,
- Tax – daň,
- Warehouse – sklad,
- WarehouseProduct – produkt na skladě.

4.3.2 Repositáře

Každá entita má v systému vytvořený svůj vlastní repositář, který rozšiřuje funkcionalitu základního „EntityRepository“. Díky tomu je možné jednoduše definovat vlastní speciální metody pro načítání entit z databáze přímo do konkrétních repositářů.

4.3.3 Kontejnery

Ke každé entitě existuje i kontejner stejně jako existuje ke každé entitě repositář. Tento kontejner je definován jako služba (jeho instanci tedy poskytuje framework prostřednictvím DI kontejneru aplikace). Kontejner jako takový usnadňuje práci tím způsobem, že sám obsahuje instance důležitých tříd pro práci s entitami a místo importu několika různých tříd se provede pouze import instance tohoto kontejneru (přes DI kontejner).

Samotný kontejner je schopen poskytnout:

- název entity ke které se vztahuje,
- instanci manažera entit (řídí načítáním ukládání a mazání dat v úložišti – databázi),
- instanci repositáře entity,
- novou instanci entity.

4.3.4 Fasády

Jelikož entita je sama o sobě pouze nosičem informací a neměla by obsahovat žádnou složitější logiku (např.: metody pro načtení všech obrázků produktu spolu se všemi obrázky všech variant produktu), byly vytvořeny fasády, ve kterých je možné definovat tyto pokročilé metody. Výhodou tohoto řešení je čisté oddělení informací uložených v entitě od pokročilých metod.

4.4 Funkcionalita systému

Systém obsahuje spoustu funkcionality. V této sekci jsou popsány ty nejdůležitější a nejvíce zajímavé.

4.4.1 Systémová oprávnění

Veškerá funkcionalita v administrační části systému a většina funkcionality v modulu Front je podmíněna příslušným oprávněním. Ty jsou spravovány v administrační části systému. Lze tak uživatelským rolím povolit oprávnění k příslušným akcím. Kontrola oprávnění je potlačena pouze v případě, že je požadavek spuštěn pomocí konzolového příkazu.

Uživatelské role

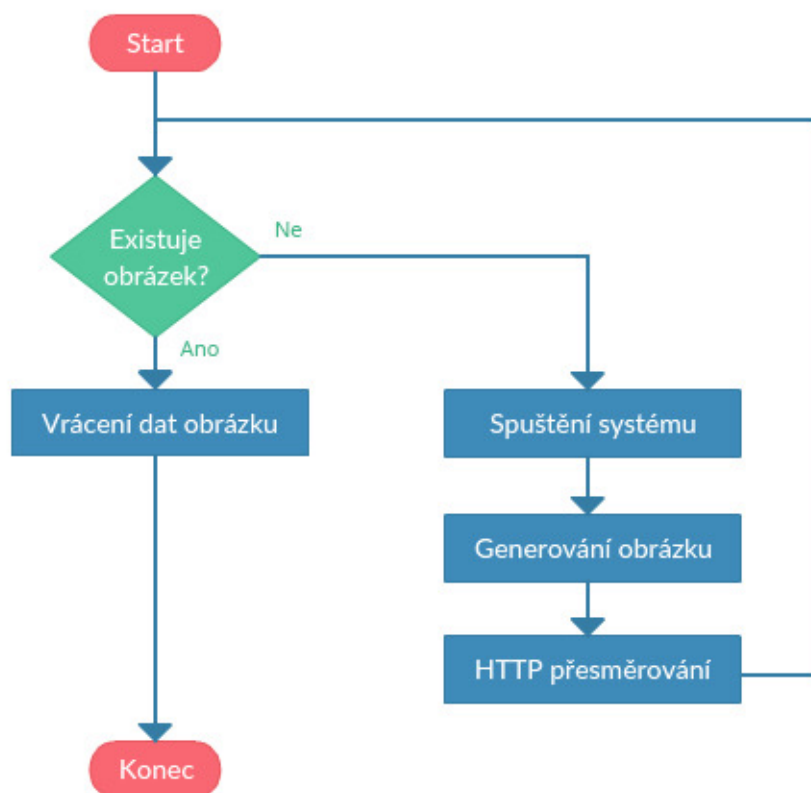
Uživatelská role slouží k definování systémových rolí uživatelů. Každému uživateli lze přiřadit právě jednu z těchto rolí.

Zdroje oprávnění

Zdroje oprávnění slouží k definici konkrétního oprávnění pro uskutečnění konkrétní akce v systému. Například oprávnění do administrační sekce, oprávnění k editaci produktů, oprávnění pro přístup do uživatelské sekce, atd.

4.4.2 Asynchronní zpracování obrázků

Pro zobrazování obrázků je používáno asynchronní zpracování. To znamená, že žádný obrázek v obsahové části systému není zobrazen v originální podobě, místo toho se zobrazí systémem zpracovaný obrázek.

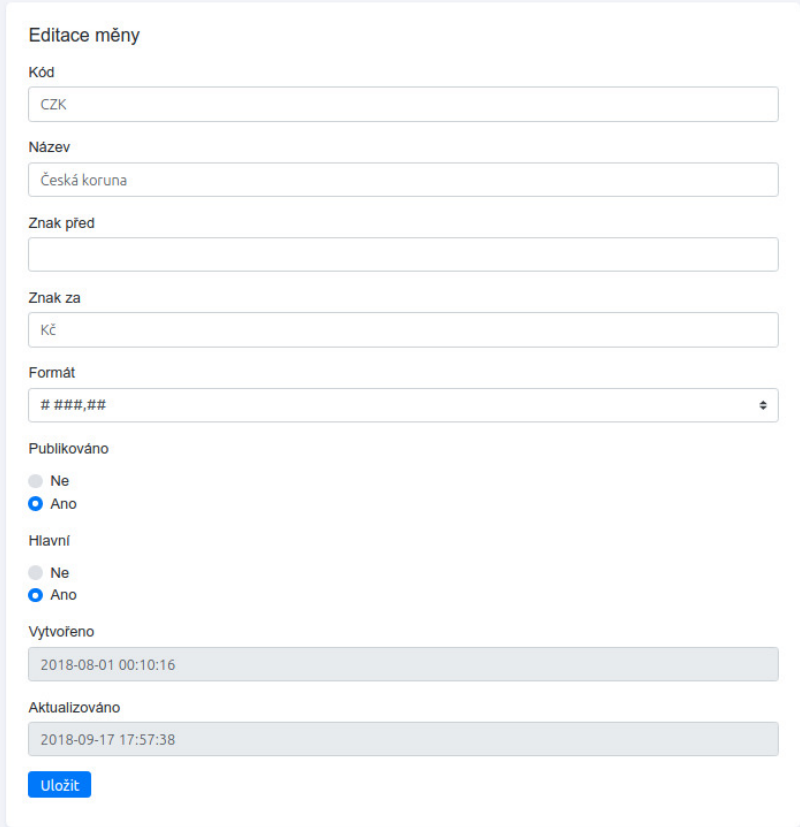


Obrázek 4.11: Vývojový diagram procesu asynchronního generování obrázků, zdroj: vlastní zpracování

Obrázek je nejprve v originální podobě nahrán na server, následně až při jeho zobrazení jej systém upraví do požadované formy. Ke zpracování obrázku pak dochází až při samotnému HTTP požadavku, kdy se prohlížeč pokusí stáhnout obsah (obrázek) na konkrétní URL, kterou vygeneroval systém. V této URL se nachází zakódované informace o tom, jak má být obrázek konkrétně zpracován. Pokud se obrázek v této formě na serveru nenachází, dojde ke spuštění úlohy systému, která obrázek upraví. Pro zpracovávání obrázků se používá standardní knihovna pro zpracovávání obrázků v Nette Framework.

4.4.3 Automatické sestavování formulářů

Vytváření formulářů pro úpravy entit je zdoluhavé a díky velkému množství entit v systému i nepříliš zábavné. Proto byl vytvořen automatický generátor formulářů entit, který formuláře pro entity automaticky sestaví.



Formulář pro editaci měny (Currency Edit) s následujícími poli:

- Kód: CZK
- Název: Česká koruna
- Znak před: (prázdné)
- Znak za: Kč
- Formát: ###,##
- Publikováno: ☐ Ne, ☒ Ano
- Hlavní: ☐ Ne, ☒ Ano
- Vytvořeno: 2018-08-01 00:10:16
- Aktualizováno: 2018-09-17 17:57:38
- Uložit (modré tlačítko)

Obrázek 4.12: Ukázka automaticky vygenerovaného formuláře pro úpravu měny, zdroj: administrační rozhraní Jarvis

Při zobrazení editace entity tedy stačí pouze nastavit, které vlastnosti entity nemají být zobrazeny, které jsou pouze pro čtení a generátor automaticky sestaví formulář pro úpravu entity. Při úpravě entity (změně její struktury) je pak formulář zcela automaticky sestaven pro aktuální strukturu entity. Logika mapování formulářů navíc dokáže namapovat entity na vygenerovaný formulář i zpět pro následnou validaci a uložení entity. Popisky polí se vytvářejí pomocí překladače, popisek každého pole tedy lze pohodlně lokalizovat.

4.4.4 Export entit do Elasticsearch

Kvůli tomu, že například pro výpis a filtrování produktů v kategoriích pro účely modulu Front je využíváno fulltextového vyhledávání pomocí Elasticsearch, je nutné entity (např. produkty) do Elasticsearch exportovat.

Toho je docíleno tím, že na akci uložení entity do databáze pomocí manažera entit (který se stará o ukládání a načítání entit z databáze) je navázána akce, kdy se spouští logika exportéru. Jeho úkolem je detekovat, zda je povoleno danou entitu exportovat, která data se v entitě změnila, zda tyto změny jsou z pohledu exportu do Elasticsearch pro další použití podstatná a případně zaregistrovat do fronty (RabbitMQ) pro zpracování exportu.



Obrázek 4.13: Export entity do Elasticsearch, zdroj: vlastní zpracování

Samotný export je následně zpracován v paralelním vlákně, kde se exportér pokusí načíst entitu z databáze a pokud se nepodaří entitu načíst, došlo k jejímu odstranění a exportér odešle požadavek na Elasticsearch na smazání indexu. Pokud dojde k načtení entity z databáze, je entita dehydratována do podoby asociativního pole a odeslána do Elasticsearch k zaindexování.

4.4.5 Import entit z Elasticsearch

Na základě vyhledávání v Elasticsearch jsou systému nazpět navracena data, které indexy splňují kritéria dotazu vyhledávání. Aby nemuselo docházet k opětovnému načtení entit z databáze, byla vytvořena logika pro hydrataci výsledků navracených z Elasticsearch do plnohodnotných entit. S těmito entitami pak lze pracovat stejně, jako kdyby tyto entity byly načteny pomocí manažera entit z databáze.



Obrázek 4.14: Import entity z Elasticsearch, zdroj: vlastní zpracování

4.4.6 Lokalizace

Lokalizace systému je realizovaná na úrovni jazyků a měn. Administrátoři mají k dispozici nadefinovat jazyk a měny, které bude systém používat a vždy zvolí jeden jazyk a jednu měnu jako hlavní (výchozí). Výchozím jazykem celého systému je ve výchozím stavu angličtina, měna ve výchozím stavu není definovaná žádná.

Pokud je pak entita přeložitelná (existuje entita překladu konkrétní entity) je formulář pro úpravu entity automaticky rozšířen o překladová pole, které překládají vlastnosti entity do konkrétního jazyku.

o kolik procent bude cena snížena oproti základní cenové hladině – v základní cenové hladině pro základní měnu musí být cena definována).

The screenshot displays the 'Ceny produktu' (Product Prices) configuration page. At the top, there is a dropdown menu labeled 'Přidat cenu pro měnu' (Add price for currency) with the option 'Vyberte' (Select). Below this, two currency sections are shown. The first section is for 'Česká koruna - CZK' (Czech Koruna). It features a 'Základní hladina' (Basic level) input field with the value '100' and a 'Kč' (CZK) currency selector, and a 'Snížená hladina' (Reduced level) input field with the value '60' and a 'Kč' currency selector. Below the 'Základní hladina' input, it says 'Základní cenová hladina' (Basic price level). Below the 'Snížená hladina' input, it says '80 Kč'. A red 'Odstranit' (Remove) button is located to the right of the 'Česká koruna - CZK' section. The second section is for 'Euro - EUR'. It features a 'Základní hladina' input field with the value '54' and a '€' (EUR) currency selector, and a 'Snížená hladina' input field with the value '20' and a '€' currency selector. Below the 'Základní hladina' input, it says 'Základní cenová hladina'. Below the 'Snížená hladina' input, it says '€ 43.2'. A red 'Odstranit' button is also present to the right of this section. At the bottom left of the form, there is a blue 'Uložit' (Save) button.

Obrázek 4.16: Konfigurace cen produktu, zdroj: administrační rozhraní Jarvis

Návštěvník má poté možnost přepnout si dostupné rozhraní do některého z nabízených jazyků a měn.

4.4.7 Vytváření URL adres

Vytváření URL adres pro moduly Admin a Robot je realizováno pomocí jednoduchého statického routeru.

Pro modul Front jsou URL adresy sestavovány za pomoci Route entit. Ty se jako zbylé entity ukládají do databáze a systém díky nim dokáže obousměrně překládat mezi URL a akcí konkrétního presenteru. To má za následek, že entitám lze definovat uživatelsky přívětivé URL adresy.

Route entity jsou generovány automaticky na pozadí při manipulacích s ostatními entitami v systému, které podporují routování (implementují rozhraní „IRoutable“).

Například, pokud je vytvořena entita Article (entita článku – podporuje routování), dojde i k automatickému vytvoření entity Route, která pomocí reference odkazuje na entitu článku. Jako URL adresa slouží název článku, který je převe-

den do podoby pro použití v URL adrese. Pokud je daný článek upraven (a změní se například jeho název, který má podstatný vliv na tvorbu URL adresy), nedojde k aktualizaci entity Route, ale je vytvořena zcela nová a stará upravena tak, že již neodkazuje na entitu článku, ale na novou Route entitu. Výsledkem této funkcionality je automatické přesměrování starších URL adres na novou (které mohou být již zaindexovány vyhledávači nebo na ně mohou odkazovat statické odkazy). Díky tomu je také zabráněno duplikaci stejného obsahu na více různých URL adresách, což má neblahý vliv na SEO.

Entity systému, které vytvářejí URL adresy:

- Article (článek),
- Page (stránka),
- Product (produkt).

Entita Page (stránka) umožňuje vytvořit buď prázdnou stránku, jejíž zobrazení se složí z výpisu konkrétních modulů nebo výpisu kategorie článků či produktů.

4.4.8 Administrační rozhraní

Administrační rozhraní systému se nachází na adrese „<doména>/administration“. Pro vstup do administrace je nutné mít vytvořený uživatelský účet, kterému je přiřazena role s dostatečným oprávněním pro vstup do administrace. Pokud uživatel není přihlášen, je mu zobrazen formulář pro přihlášení.



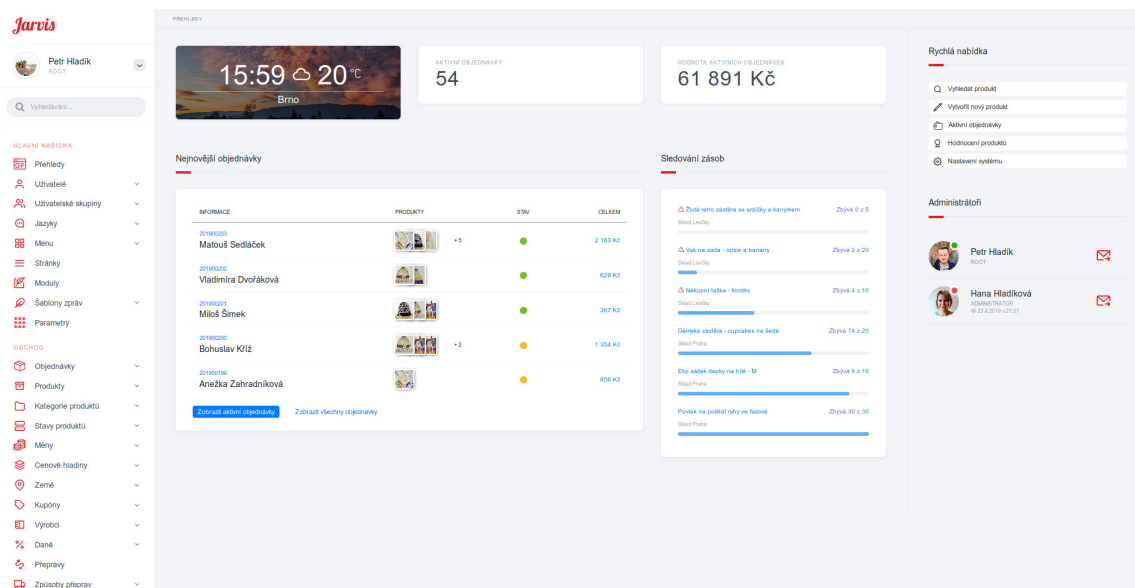
Obrázek 4.17: Přihlášení do administračního rozhraní systému, zdroj: administrační rozhraní Jarvis

Pro zpestření zobrazení přihlašovací stránky je při každém jejím zobrazení načten náhodný obrázek za pomoci veřejně dostupné služby Unsplash Source.



Obrázek 4.18: Přihlášení do administračního rozhraní systému po změně obrázku, zdroj: administrační rozhraní Jarvis

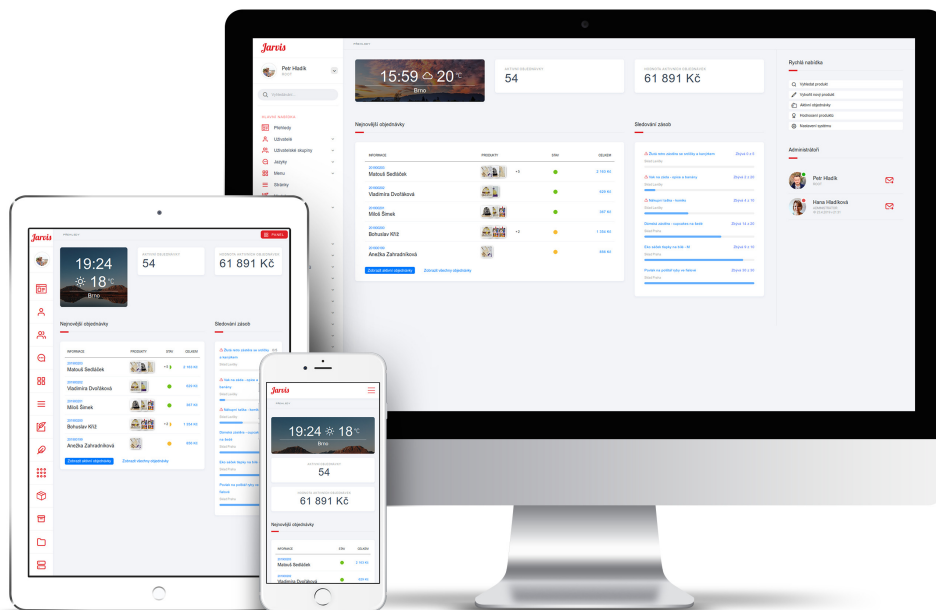
Po přihlášení je uživatel přesměrován již do plnohodnotného uživatelského rozhraní, konkrétně na nástěnku systému, kde vidí sumarizaci nejdůležitějších informací.



Obrázek 4.19: Nástěnka administračního rozhraní systému, zdroj: administrační rozhraní Jarvis

Uživatel má v administraci možnost administrovat veškeré entity, které obsahuje systém a ke kterým má oprávnění editace. Může tedy například administrovat produkty, objednávky, uživatele, a další entity.

Celé administrační rozhraní je navrženo tak, aby bylo pohodlně administrovatelné například z mobilního telefonu nebo tabletu – šablona administrace je tedy plně responzivní.



Obrázek 4.20: Nástěnka administračního rozhraní systému, zdroj: vlastní zpracování

4.4.9 Uživatelské parametry

V administrační sekci lze definovat uživatelské parametry, které reprezentují konkrétní vlastnosti entit v systému a tyto vlastnosti těmto entitám přiřadit. Každý parametr musí být definovaný konkrétním typem, přičemž na výběr je z možností:

- číslo (number),
- textový řetězec (string),
- výběr stavu pravda / nepravda (boolean),
- výběr jedné z hodnot (select),
- výběr více hodnot (multiselect).

Takto lze například entitě produktu definovat vlastnost „velikost“ s typem výběru jedné z hodnot „S“, „M“, „L“.

Přehled parametrů

ID	NÁZEV	UID	TYP	AKCE	
6	Velikost	size	select	Editovat	Odstranit
5	Typ	type	string	Editovat	Odstranit
4	Počet kapes	numberofpockets	number	Editovat	Odstranit
3	Materiál	material	multiselect	Editovat	Odstranit
2	Barva	color	select	Editovat	Odstranit
1	Extra velikost	extrasize	bool	Editovat	Odstranit

(Položky: 1 - 6 z 6) 20

Obrázek 4.21: Ukázka výpisu definovaných uživatelských parametrů, zdroj: administrační rozhraní Jarvis

Tato funkcionality má obrovskou výhodu při customizaci řešení, kdy konkrétní úpravy ve vlastnostech entit dle potřeb zákazníka se nemusí řešit na úrovni úpravy zdrojového kódu základního modelu entity, ale postačí vhodně nakonfigurovat uživatelské parametry a dle potřeby upravit například pouze šablonu pro jejich speciální výpis.

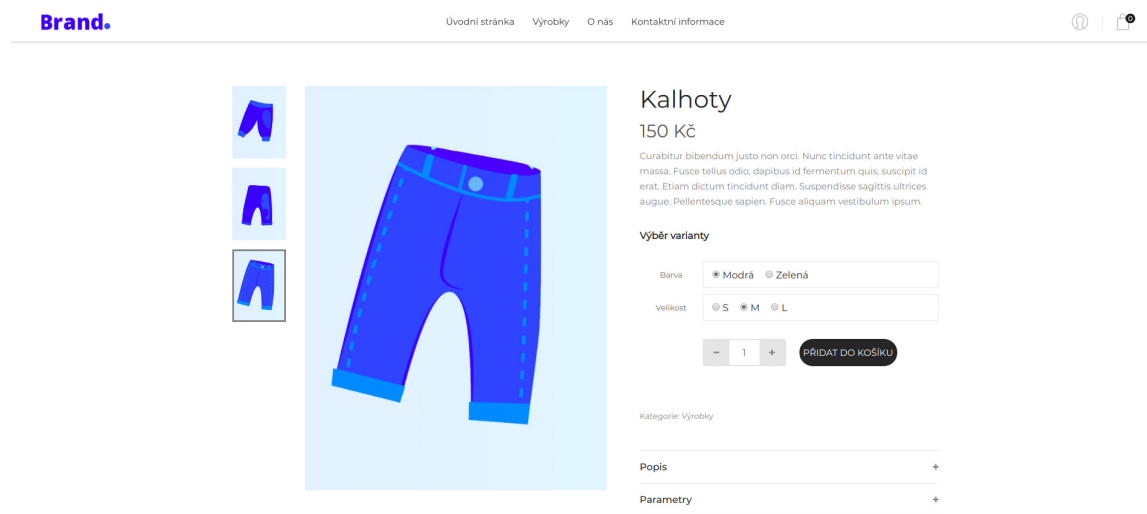
Navíc této funkcionality bylo využito i při výpisu variant produktů, kdy jsou jednotlivé varianty od sebe primárně odlišeny v rozdílných hodnotách uživatelských parametrů (vlastnostech).

4.4.10 Varianty produktů

Ke každému produktu mohou být definovány jeho varianty. Ty jsou reprezentovány pomocí samostatných entit produktů, které ale odkazují na entitu hlavního produktu jako na svého rodiče.

To má svoji výhodu v tom, že takováto varianta může obsahovat svoje vlastní obrázky, parametry, dostupnost i ceny stejně, jako plnohodnotný produkt a na tuto variantu lze pak odkazovat stejně, jako na produkt.

Jednotlivé varianty produktů se pro zákaznický výběr detekují na základě odlišností v hodnotách uživatelských parametrů.



Obrázek 4.22: Ukázka výběru varianty produktu ve výchozí šabloně detailu produktu, zdroj: systém Jarvis

4.4.11 Customizace systému

Customizaci, neboli specifická úprava systému dle přání zákazníka, lze provádět v systému dvojím způsobem:

- na úrovni přepisování služeb,
- na úrovni přepisování šablon.

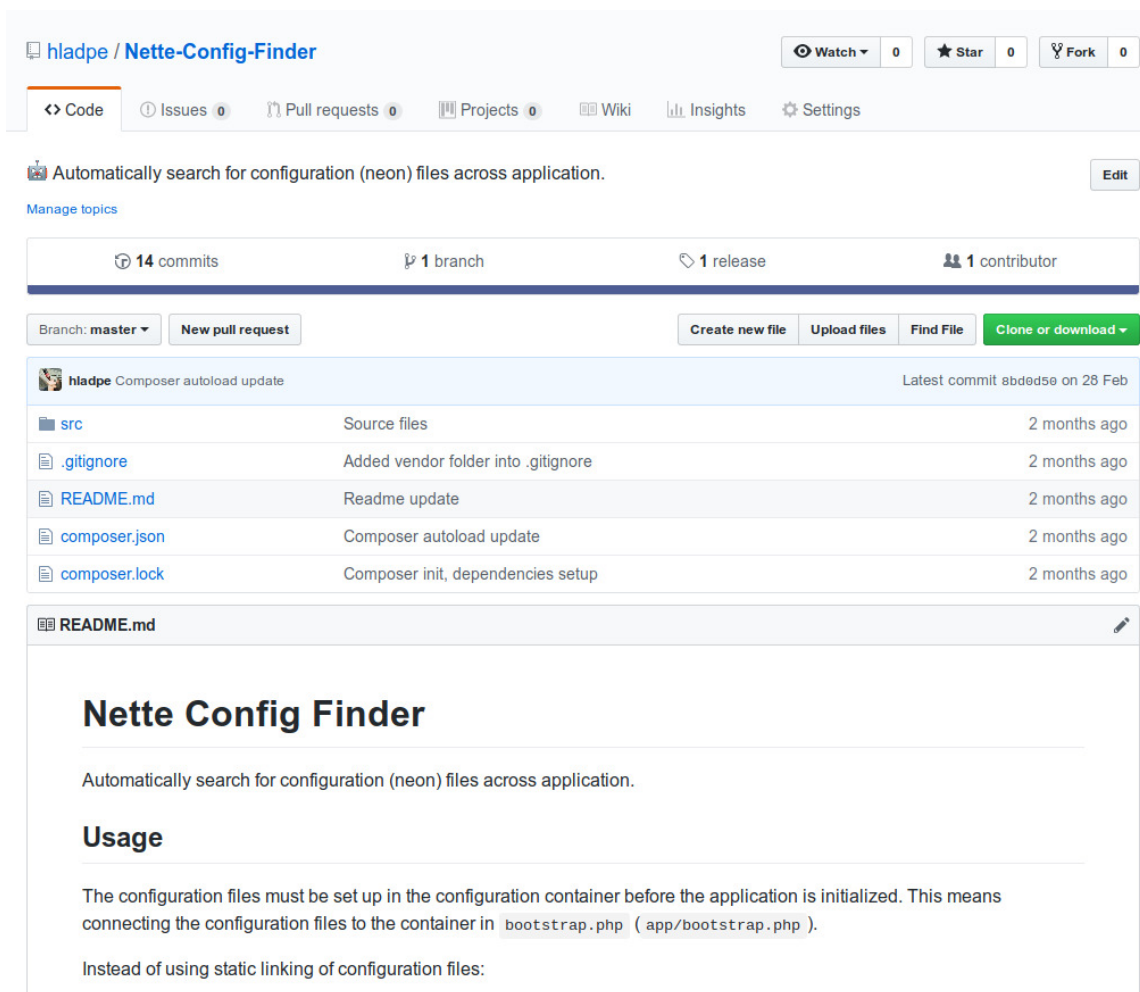
Ani jeden z uvedených způsobů nezasahuje do systému tak, že by přímo přepisoval zdrojový kód samotného systému, ale jsou v obou případech vytvořeny jako externí soubory oddělené od adresářů zdrojových kódů systému, ale ten je automaticky načítá a pracuje s nimi.

4.4.12 Přepisování služeb

Nette Framework umožňuje pomocí konfiguračních souborů NEON přepisovat již dříve definované konfigurace. V systému konfigurační soubory vyhledává knihovna

Nette Config Finder, která je nastavena tak, aby nejprve vyhledávala konfigurační soubory v adresáři „vendor/“, tedy aby načetla konfigurační soubory nainstalovaných balíčků a teprve poté vyhledá konfigurační soubory v adresáři „app/“.

Autorem tohoto rozšíření jsem já sám, je publikováno jako open-source a lze jej nainstalovat pomocí nástroje Composer.



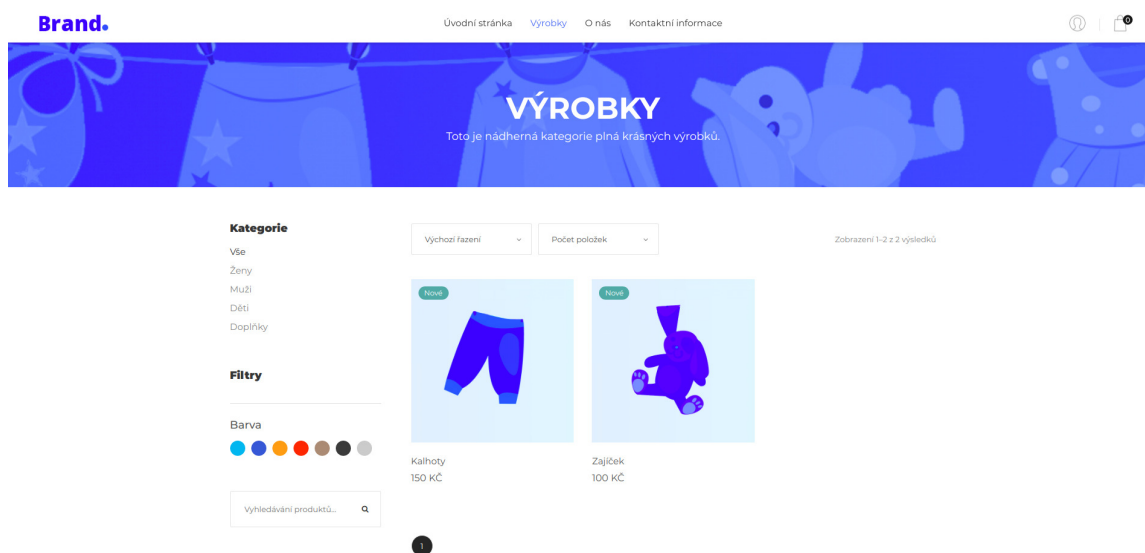
Obrázek 4.23: Open-source rozšíření Nette Config Finder, zdroj: github.com

Automatické vyhledávání konfiguračních souborů a jejich import do systému má za následek, že při inicializaci systému ve spouštěcím souboru je nejprve načtena výchozí konfigurace systému z balíčku Jarvis Core (a dalších balíčků s konfigurací) a teprve následně je načtena konfigurace z adresáře „app/config/“, tedy konkrétní konfigurace pro aktuální instanci systému.

Konfigurace z adresáře „app/config/“, přepisuje základní konfiguraci nainstalovaných balíčků a díky tomu lze donutit systém využít mimo konfiguračních parametrů i jiné služby, než které standardně systém používá. Je tedy možné měnit základní logiky systému.

4.4.13 Přepisování šablon

Komponentový model systému je rozšířen o logiku vyhledání šablony pro konkrétní komponentu. Tato logika se nejprve pokusí vyhledat šablonu konkrétní komponenty v adresáři „app/“ na základě předem jasně definovaného tvaru cesty k náhradní šabloně a pokud není nalezena náhradní šablona, využije se výchozí systémová.



Obrázek 4.24: Ukázka výchozí šablony výpisu produktů, zdroj: systém Jarvis

Tato funkcionality byla vytvořena hlavně z toho důvodu, aby samotný systém obsahoval veškeré potřebné šablony a byl díky tomu samostatně provozuschopný bez dodání jakýchkoliv dalších součástí a aby vizuální customizace systému neprobíhala formou přepisu systémových šablon.

4.4.14 Moduly

Moduly slouží pro zobrazení libovolných funkčních bloků v modulu Front, tedy v sekci pro zákazníky obchodu. Jejich správa probíhá v administraci systému a lze pomocí nich například zajistit výpis menu, obsahu košíku, libovolného článku a to na jakékoliv předem definované pozici pro vykreslení modulu. Moduly je možné ošetřit dalšími podmínkami, například, zda je povolen výpis modulu na detailu konkrétního produktu, článku nebo stránky. Hlavní myšlenkou funkcionality modulů je usnadnění práce s modifikací stránek. Například, pokud se obchod rozhodne, že potřebuje dovolenou, nemusí se upravovat nic ve zdrojovém kódu systému, ale pouze se pro tuto informaci vytvoří modul s příslušnou informací a publikuje na dané pozici.



Obrázek 4.25: Modul s informací o plánované inventuře, zdroj: systém Jarvis

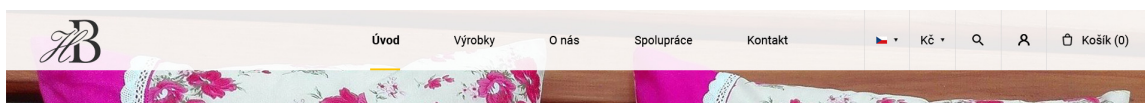
4.5 Redesign webu

Jeden z živnostníkůvých požadavků na nové řešení byla inovace vizuálního vzhledu e-shopu. Po upřesnění grafických požadavků byly vytvořeny grafické návrhy pro připravované řešení.

4.5.1 Úvodní stránka

Úvodní stránka je velmi důležitým prvkem e-shopu. Měly by se na ní objevit ty nej důležitější informace, které můžeme zákazníkům poskytnout. Grafický návrh úvodní stránky byl vytvořen tak, že prvním prvkem je hlavička stránek. Ta obsahuje logo e-shopu, výpis hlavního menu, odkaz pro přihlášení uživatele a košík. Hlavička si drží

fixní pozici vůči oknu prohlížeče a při pohybu na stránce dolů ji tedy bude návštěvník mít stále na očích.



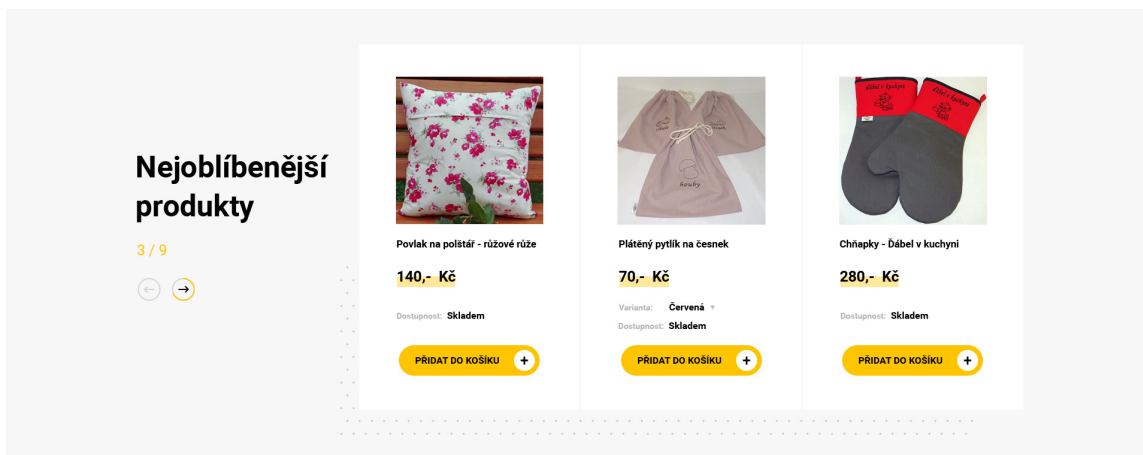
Obrázek 4.26: Hlavička webu, zdroj: vlastní zpracování

Hned pod menu se na nachází prezentace skupiny vybraných produktů. Měly by to být ty absolutně nejprodávanější, protože je zákazník vidí jako úplně první a proto by se zde měly nacházet právě ty nejoblíbenější produkty. Tuto část je vhodné animovat, například přechodem mezi prezentacemi jednotlivých produktů a vložením efektní fotky na pozadí prezentace, abychom v návštěvníkovi evokovali pocit živosti celého webu.



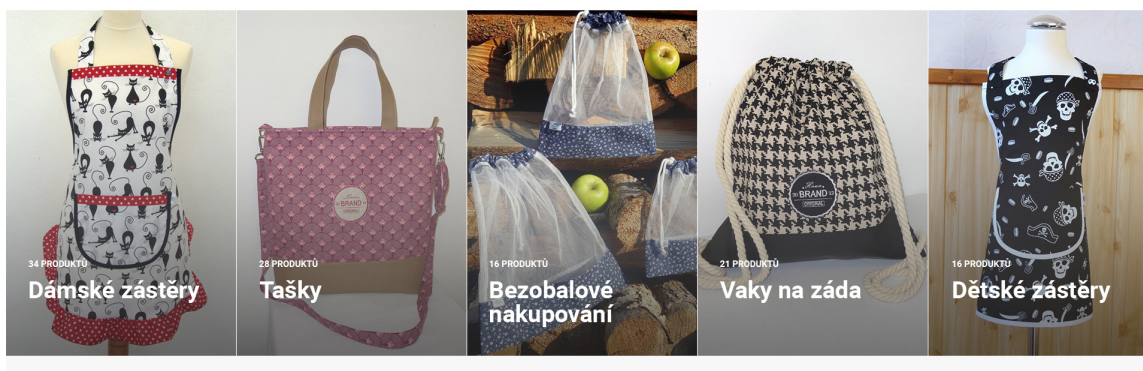
Obrázek 4.27: Produktová prezentace na úvodní stránce, zdroj: vlastní zpracování

Protože se do prezentace vejde pouze omezené množství produktů, byla vytvořena samostatná sekce nejoblíbenějších produktů ještě pod prezentací. Zde se může zobrazit zbytek oblíbených produktů, dle grafického návrhu například právě doporučených 9. Jejich zobrazení bude probíhat také formou prezentace po 3 produktech v řádku, ale tato prezentace již nebude tak bohatá na efekty, jak tomu je u hlavní prezentace. Zároveň bude mít návštěvník v tomto zobrazení produktů možnost listovat.



Obrázek 4.28: Produktová prezentace nejoblíbenějších produktů, zdroj: vlastní zpracování

Následuje výpis vybraných referencí produktů od zákazníků. Je vhodné, aby každá reference obsahovala i vlastní fotografii, ta se může použít například přímo od zákazníka nebo využít fotografie produktu, pokud reference neobsahuje vlastní.



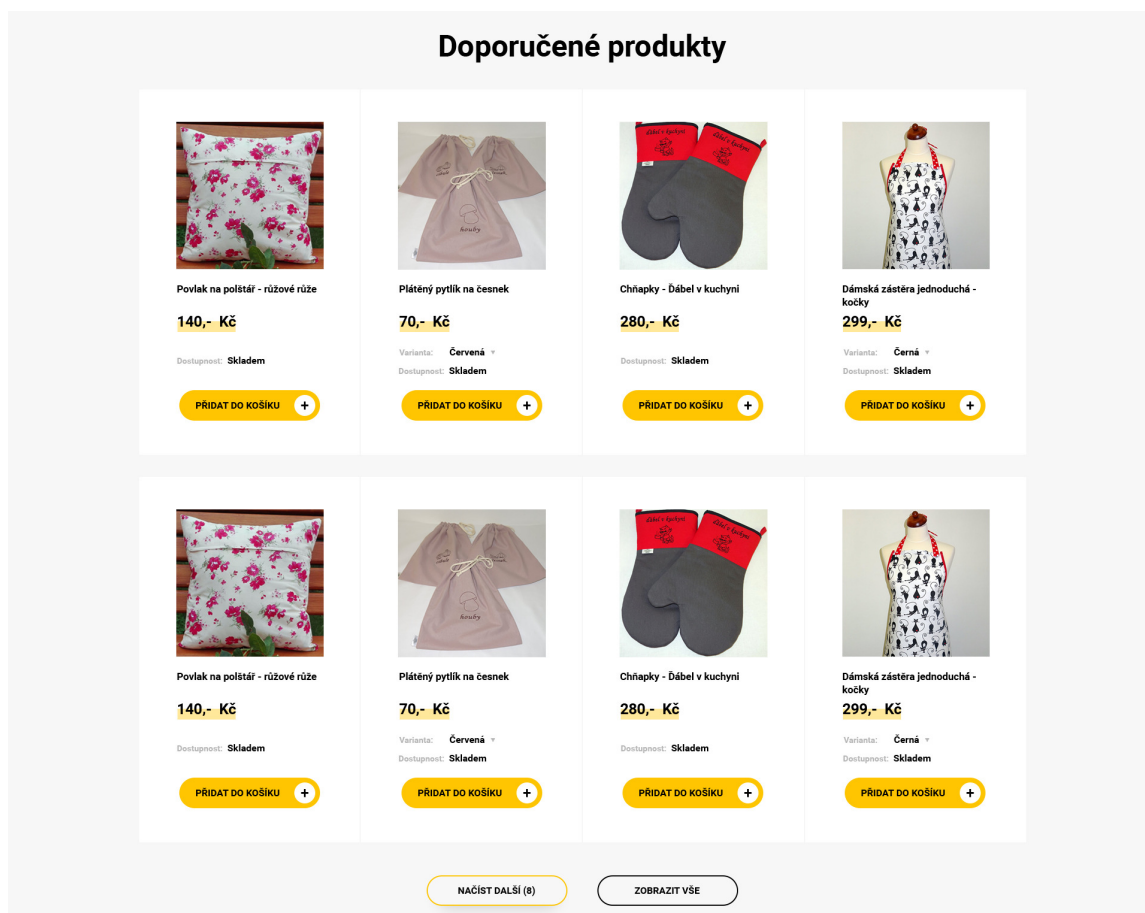
Obrázek 4.29: Výpis produktových kategorií, zdroj: vlastní zpracování

Pod výpisem referencí se nachází výpis oblíbených kategorií e-shopu, přičemž každá kategorie obsahuje vlastní fotografii.



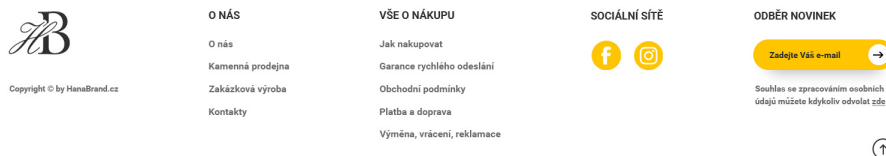
Obrázek 4.30: Výpis vybraných referencí, zdroj: vlastní zpracování

Pokud návštěvník pokračuje i pod výpis nejoblíbenějších kategorií, pravděpodobně není dostatečně přesvědčený k zobrazení detailu zobrazeného produktu či kategorie, a nebo míří k informacím uvedeným v patičce. Z toho důvodu je v grafickém návrhu ještě jednou vytvořen výpis dalších oblíbených produktů, aby se zvýšila šance na motivaci návštěvníka k zobrazení detailu některého ze zobrazených produktů.



Obrázek 4.31: Produktová prezentace oblíbených produktů, zdroj: vlastní zpracování

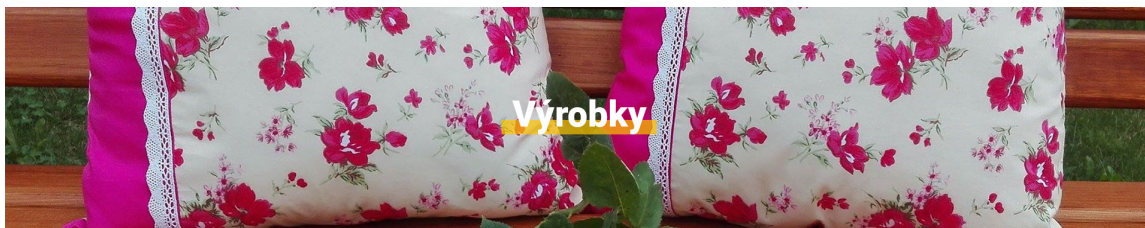
V nejspodnější části stránky se nachází výpis patičky webu. Ta obsahuje logo e-shopu, výpis menu, odkazy na sociální sítě a formulář pro přihlášení k odběru novinek.



Obrázek 4.32: Patička webu, zdroj: vlastní zpracování

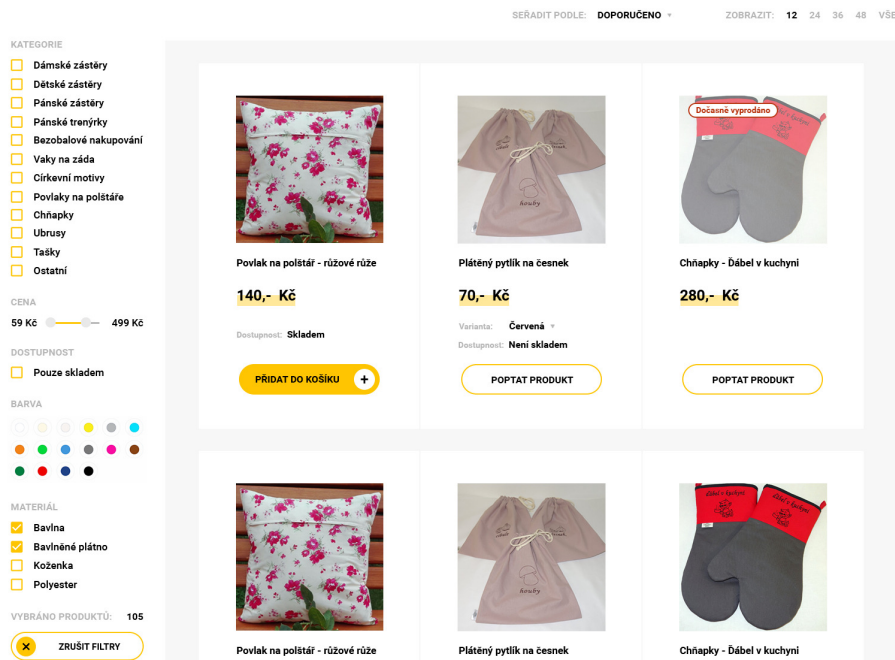
4.5.2 Detail kategorie

Detail výpisu produktové kategorie obsahuje stejnou hlavičku a patičku jako úvodní stránka. Místo prezentace nejoblíbenějších produktů je ale zobrazena lišta s názvem kategorie a fotografií kategorie na jejím pozadí. To dodá stránce opět pocit živosti, jinak by totiž byla stránka příliš jednotvárná.



Obrázek 4.33: Hlavička kategorie produktů, zdroj: vlastní zpracování

Následně je celá stránka rozdělena na dva sloupce. V tom levém menším a na první pohled méně patrném se nachází výpis podkategorií a filtrů produktů. Naopak v tom pravém větším se nachází výpis nalezených produktů v dané kategorii, případně ty, které splňují filtrační podmínky. V grafickém návrhu jsou naznačeny jak dostupné, tak nedostupné produkty. Nad samotným výpisem produktů se nacházejí parametry ovlivňující způsob výpisu produktů, tedy řazení produktů a limit počet zobrazených produktů.

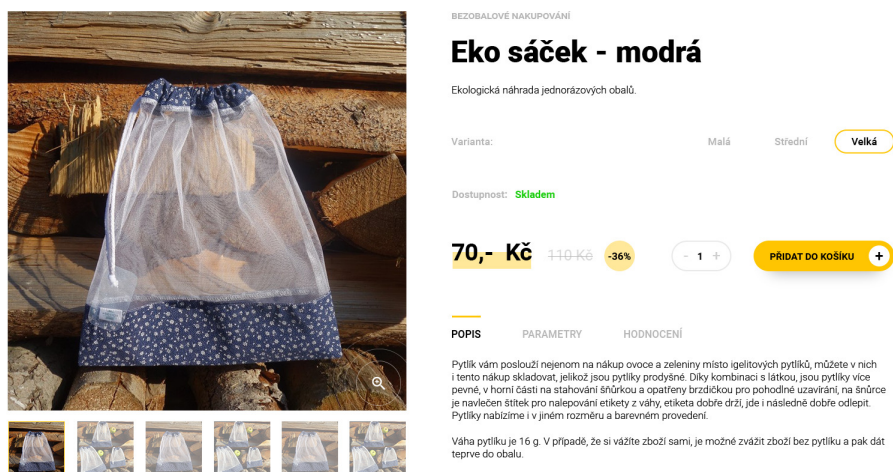


Obrázek 4.34: Výpis produktové kategorie s filtry a podkategoriemi, zdroj: vlastní zpracování

4.5.3 Detail produktu

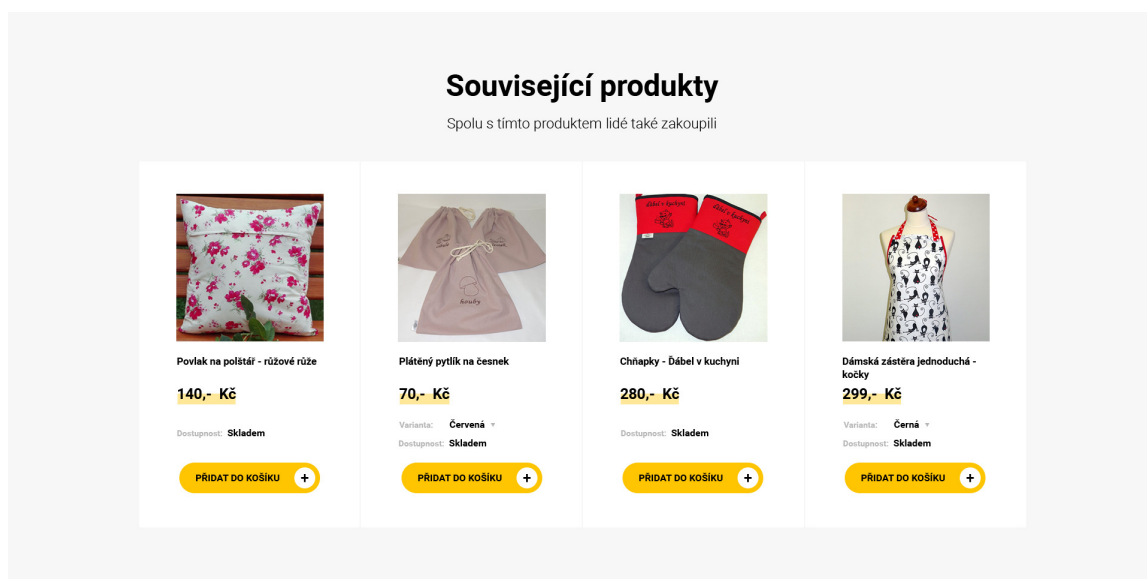
Grafický návrh detailu konkrétního produktu obsahuje opět stejnou hlavičku a patičku jako úvodní stránka. Hned pod hlavičkou webu se nachází výpis informací o daném produktu. Ten je rozdělený na dvě přibližně stejně velké části. V levé části se nachází fotografie produktu i s naznačenou ikonou pro zvětšení fotografie do plné velikosti. V pravé části se nachází možnost vložení produktu do košíku s výpisem textových informací o produktu, konkrétně:

- název,
- cena,
- výběr varianty,
- popis,
- parametry,
- hodnocení.



Obrázek 4.35: Hlavní informace o produktu, zdroj: vlastní zpracování

Pod výpisem hlavních informací o produktu je výpis souvisejících produktů. Tím je myšlen například výpis produktů ze stejné kategorie nebo výpis produktů, které zákazníci zakoupili spolu s tímto produktem.

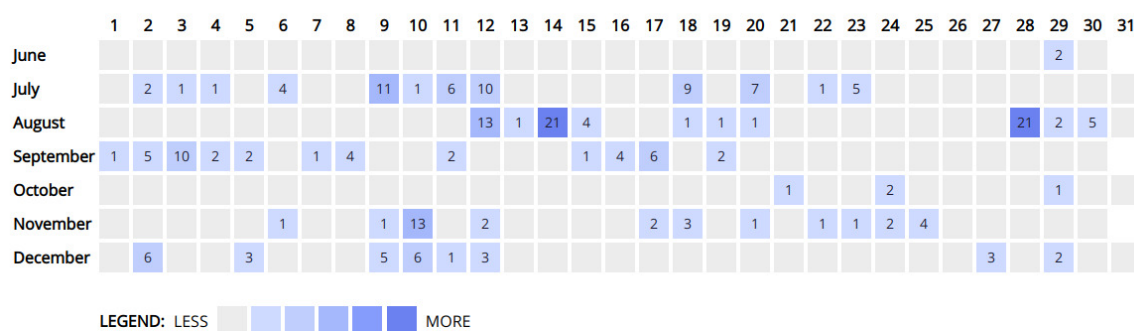


Obrázek 4.36: Výpis souvisejících produktů, zdroj: vlastní zpracování

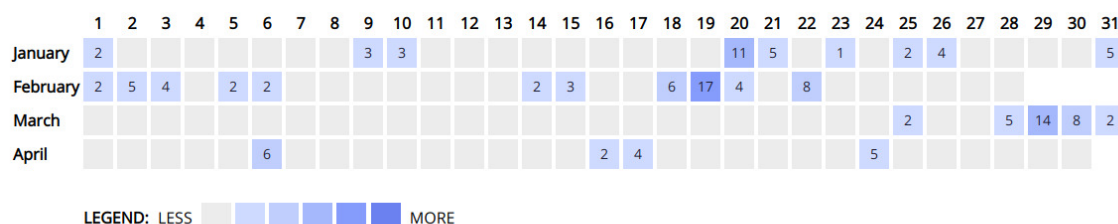
4.6 Zhodnocení realizace

Dle historie repositářů systému byly první změny nahrány dne 29.6.2018 a od té doby bylo vytvořeno celkem 373 změn.

2018



2019



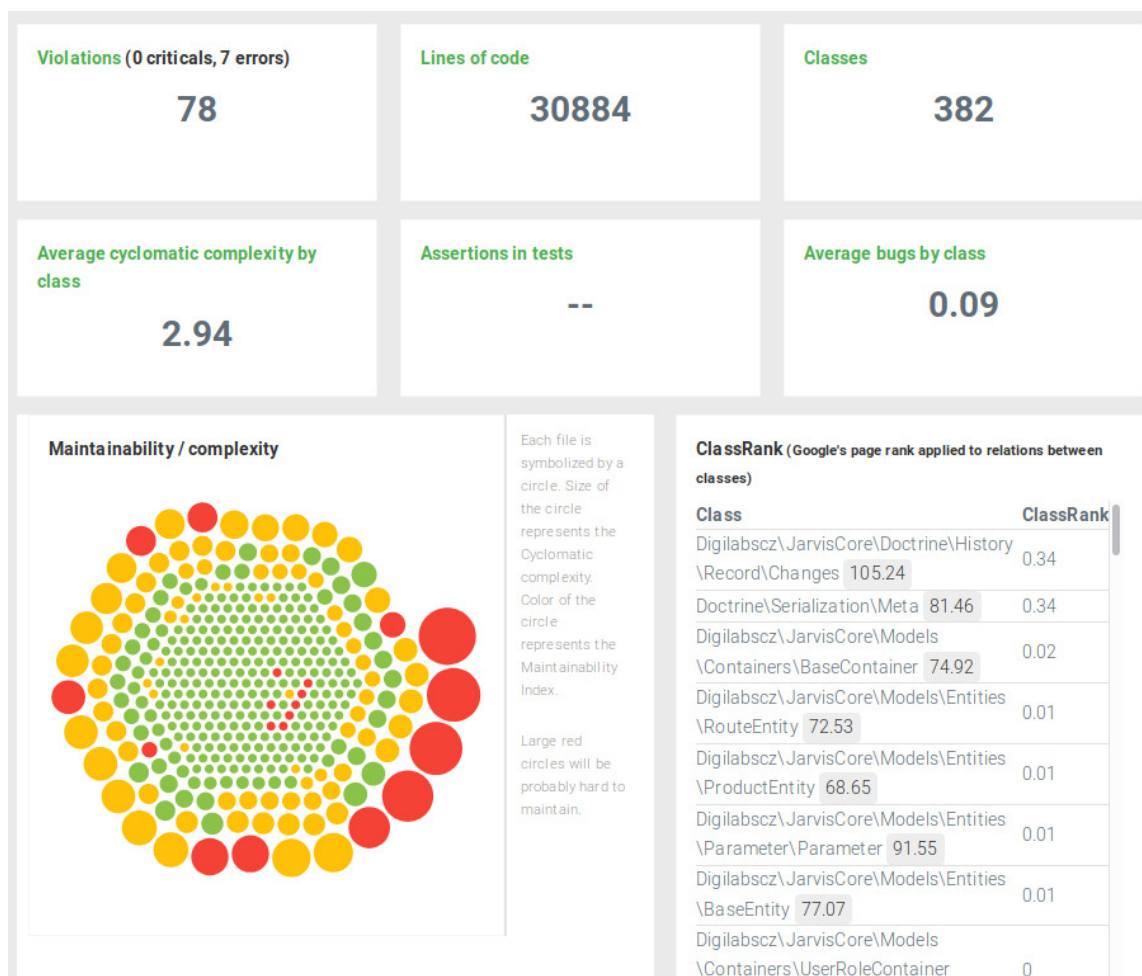
TOTAL: 373 commits

Obrázek 4.37: Historie změn v repositářích systému, zdroj: nástroj Contribution Summarizer

Odhadem se projekt nachází zhruba v polovině a konstatovat, že jeho současná podoba tvoří základ pro vytvoření plnohodnotné e-commerce platformy. V současném stavu se tedy jedná zatím o prototyp této platformy.

4.6.1 Statická analýza kódu

Nad zdrojovými kódy vytvořené platformy systému byla provedena statická analýza kódu jako nad současně používaným řešením.



Obrázek 4.38: Statická analýza zdrojových kódů systému Jarvis, zdroj: nástroj PhpMetrics

Po porovnání výsledků testu s výsledky testu aktuálně používaného řešení, je patrné, že se přibližně dvojnásobně navýšil počet řádků ve zdrojovém kódu systému a to na počet 30 884. Také se zhruba 5,5× navýšil celkový počet všech tříd, konkrétně z počtu 68 na 382.

Mírně se zvýšil počet varování na prohřešky v kódu, konkrétně z počtu 67 na 78, ale díky vyššímu počtu tříd klesl celkový průměrný počet chyb na třídu z 0,87 na 0,09. Také se snížil průměr hodnoty cyklomatické komplexnosti na třídu a to z 25,15 na hodnotu 2,94.

Z porovnání výsledků testů je patrné, že kvalita zdrojového kódu připravovaného řešení vzhledem ke svému rozsahu je výrazně vyšší, než kvalita zdrojového kódu současně používaného řešení.

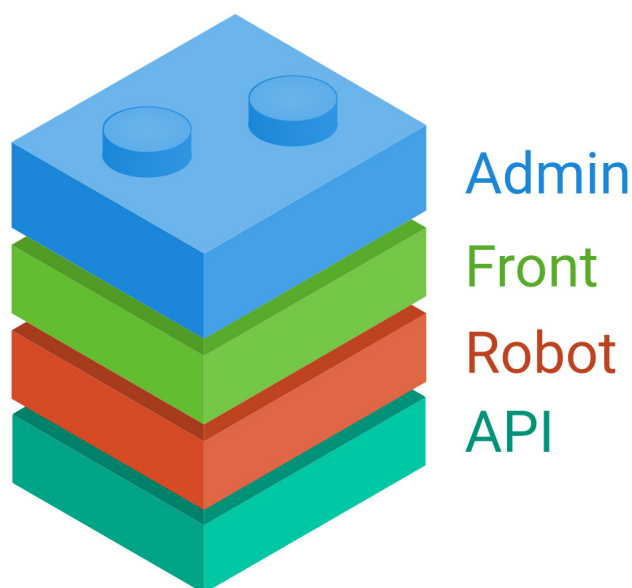
Očekává se, že v průběhu další práce na projektu se zcela eliminují nedostatky objevené těmito testy. Bylo by vhodné tyto testy provádět častěji a postupem realizace nesnižovat kvalitu zdrojového kódu systému a snažit se kvalitu spíše zvyšovat.

4.6.2 Plánovaný rozvoj

Protože celý projekt se nachází zhruba v polovině svého rozsahu, bude nutné naplánovat další kroky k dokončení celé realizace pro vytvoření plnohodnotného e-commerce řešení.

Namátkou těmito kroky jsou:

- eliminovat většinu varování ze statické analýzy kódu,
- dokončit business logiku nákupního procesu,
- vytvořit business logiku rozšíření (pluginů) systému,
- dokončit realizaci všech požadovaných funkcí od živnostníka,
- vytvořit exportní a importní funkcionality,
- vytvořit 4. modul pro správu systému pomocí API,
- provést bezpečnostní audit systému a eliminovat případné nedostatky,
- a další.



Obrázek 4.39: 4 Moduly systému, zdroj: vlastní zpracování

Po dokončení realizace celého e-commerce řešení bude také nutné realizovat potřebné kroky k samotnému zprovoznění řešení pro živnostníka. Bude tedy například nutná instalace systému na vhodných server (tedy najít vhodného poskytovatele hostingových služeb), nakonfigurovat systém, customizovat ho (přípravit a nastavit šablony zákaznické sekce e-shopu), a provést migrace dat ze současného řešení. Před samotným spuštěním bude samozřejmě nutné, aby bylo řešení dostatečně otestováno i samotným živnostníkem.

4.6.3 Finanční zhodnocení

Jak bylo již zmíněno, k prvnímu nahrání změn do repositářů systému došlo 29.6.2018. Od té doby bylo vytvořeno celkem 373 změn a realizací jsem strávil přibližně 231 hodin.

Jelikož celé řešení je postaveno na technologiích, které jsou nabízeny zdarma, nebo jako open-source, jedinou nákladovou položkou v realizaci je strávený čas nad projektem. Pokud tedy budeme uvažovat o sazbě 500 Kč/h, současné náklady na vytvoření tohoto prototypu e-commerce platformy jsou 115 500 Kč.

Připravované řešení je v porovnání s aktuálním výrazně komplexnější a díky tomu má i vyšší nároky na prostředí, ve kterém bude provozován. Kvůli tomu bude nutné při zprovoznění řešení uvažovat i o dodatečných nákladech za využívané hostingové služby (pronájem výkonnějšího serveru).

Kapitola 5

Závěr

Cílem práce bylo navrhnout a vytvořit prototyp nové e-commerce platformy, která bude v budoucnu tvořit základ plnohodnotného řešení pro konkrétního živnostníka. Toto připravované řešení bude mít za úkol zcela nahradit jeho současně používané e-shopové řešení a to tak, že bude splňovat veškeré živnostníkovy technické požadavky.

V rámci práci byly nejprve definovány nutné obecné předpoklady pro pochopení problematiky, včetně seznámení s aktuálně používaným e-shopovým řešením a analýzou jeho současného technického stavu. Následně byly definovány živnostníkovy klíčové požadavky na připravované řešení a na základě nich byla provedena analýza dostupných řešení.

V kapitole zabývající se samotným řešením jsou obsaženy informace o zvolených technologiích včetně popisu, jak přesně byly tyto technologie v projektu využity. Konkrétně se kapitola zabývá informacemi o způsobu implementace zvolených technologií, strukturou celého systému (způsobu rozdělení na samostatné balíčky a aplikační moduly), modelovou vrstvou systému, popisem konkrétních funkcionalit, grafickým návrhem připravovaného e-shopu a zhodnocením celé realizace včetně finančního zhodnocení.

Výsledkem této práce je funkční prototyp e-commerce platformy s plánovaným vývojem v plnohodnotné řešení.

Literatura

- [1] Co je internet a jak funguje? - datacentrum wedos. *Datacentrum.wedos.com* [online]. 2010 [cit. 2019-05-02]. Dostupné z: <https://datacentrum.wedos.com/a/17/co-je-internet-jak-funguje.html>.
- [2] Historie internetu. *Home.zcu.cz* [online]. 2013 [cit. 2019-04-11]. Dostupné z: <https://home.zcu.cz/tisarova/>.
- [3] Jiří Sedláček. *E-komerce, internetový a mobil marketing od A do Z*. BEN - technická literatura, Praha, 1. vyd. edition, 2006.
- [4] 3.2.5 služby internetu. *Is.muni.cz* [online]. [cit. 2019-04-26]. Dostupné z: <https://is.muni.cz/do/ics/el/sitmu/law/html/sluzby-internetu.html>.
- [5] Instant messaging – wikiknihovna. *Wiki.knihovna.cz* [online]. [cit. 2019-04-10]. Dostupné z: http://wiki.knihovna.cz/index.php/Instant_messaging.
- [6] Sociální sítě – živě.cz. *Zive.cz* [online]. 2010 [cit. 2019-05-05]. Dostupné z: <https://www.zive.cz/socialni-site/sc-223/default.aspx>.
- [7] Internetový prohlížeč. *Helpmark.cz* [online]. 2013 [cit. 2019-04-15]. Dostupné z: <https://www.helpmark.cz/slovníkpojmu/33-internetovy-prohlizec>.
- [8] Webová stránka (internetová stránka) - managementmania.com. *Managementmania.com* [online]. 2012 [cit. 2019-04-25]. Dostupné z: <https://managementmania.com/cs/webova-stranka-internetova>.
- [9] Definice statické webové stránky. *Myartse.com* [online]. 2014 [cit. 2016-04-01]. Dostupné z: <http://www.myartse.com/definice-staticke-webove-stranky/>.
- [10] Co je to hypertext? - it slovník. *It-slovník.cz* [online]. 2013 [cit. 2019-04-16]. Dostupné z: <https://it-slovník.cz/pojem/hypertext>.
- [11] Petr Suchánek. *E-commerce: elektronické podnikání a koncepce elektronického obchodování*. Ekopress, Praha, 1. vyd. edition, 2012.
- [12] Co je e-commerce | adaptic. *Adaptic.cz* [online]. 2010 [cit. 2019-04-12]. Dostupné z: <http://www.adaptic.cz/znalosti/slovníček/e-commerce/>.
- [13] K.C. Laudon and C.G. Traver. *E-commerce: Business, Technology, Society*. Pearson, 2016.

- [14] Co je e-shop | adaptic. *Adaptic.cz* [online]. 2011 [cit. 2019-04-18]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/e-shop/>.
- [15] E-commerce - managementmania.com. *Managementmania.com* [online]. 2012 [cit. 2019-04-10]. Dostupné z: <https://managementmania.com/cs/e-commerce>.
- [16] B2b (business to business) - managementmania.com. *Managementmania.com* [online]. 2012 [cit. 2019-04-25]. Dostupné z: <https://managementmania.com/cs/business-to-business>.
- [17] Co jsou zkratky b2c, b2b, b2g, b2e | altaxo se. *Altaxo.cz* [online]. [cit. 2019-04-06]. Dostupné z: <https://www.altaxo.cz/provoz-firmy/marketing/co-jsou-zkratky-b2c-b2b-b2g-b2e>.
- [18] Co to je b2b, b2c, b2g, c2b, b2a, b2e, b2r, c2c, c2g, g2b, g2c, g2g... | mladypodnikatel.cz. *Mladypodnikatel.cz* [online]. 2012 [cit. 2019-04-13]. Dostupné z: <https://mladypodnikatel.cz/b2b-b2c-b2g-c2b-b2a-b2e-b2r-c2c-c2g-g2b-g2c-g2g-t950>.
- [19] B2c (business to customer) - managementmania.com. *Managementmania.com* [online]. 2012 [cit. 2019-05-03]. Dostupné z: <https://managementmania.com/cs/business-to-customer>.
- [20] C2b (customer to business) - managementmania.com. *Managementmania.com* [online]. 2012 [cit. 2019-05-01]. Dostupné z: <https://managementmania.com/cs/customer-to-business>.
- [21] C2c (customer to customer) - managementmania.com. *Managementmania.com* [online]. 2012 [cit. 2019-04-30]. Dostupné z: <https://managementmania.com/cs/customer-to-customer>.
- [22] E-commerce: světový obchod online - edotace. *Edotace.cz* [online]. 2017 [cit. 2019-04-08]. Dostupné z: <http://www.edotace.cz/clanky/e-commerce-svetovy-obchod-online>.
- [23] Přečtěte si hlavní výhody a nevýhody e-shopů, jak je vidí jejich provozovatelé - podnikatel.cz. *Podnikatel.cz* [online]. 2017 [cit. 2019-05-05]. Dostupné z: <https://www.podnikatel.cz/clanky/prectete-si-hlavni-vyhody-a-nevyhody-e-shopu-jak-je-vidi-jejich-provozovatele/>.
- [24] Mirek Sedlák. *Jak vytvořit úspěšný a výdělečný internetový obchod*. Computer Press, Brno, 1. vyd. edition, 2012.
- [25] Open-source řešení pro e-shop – kdy ho zvolit a jak se při výběru nespálit? | mladypodnikatel.cz. *Mladypodnikatel.cz* [online]. [cit. 2019-05-04]. Dostupné z: <https://mladypodnikatel.cz/open-source-reseni-pro-e-shop-kdy-ho-zvolit-a-jak-se-pri-vyberu-nespalit-t33530>.

- [26] Tvorba obchodů na míru - shoptet.cz. *Shoptet.cz* [online]. 2011 [cit. 2019-04-29]. Dostupné z: <https://www.shoptet.cz/tvorba-e-shopu-na-miru/>.
- [27] Proč si (ne)pořídít e-shop na míru? - desítka.cz. *Desítka.cz* [online]. [cit. 2019-04-09]. Dostupné z: <https://www.desitka.cz/text-proc-si-ne-poridit-e-shop-na-miru/>.
- [28] Co je online marketing | mioweb slovníček webových pojmů. *Mioweb.cz* [online]. [cit. 2019-05-04]. Dostupné z: <https://www.mioweb.cz/slovnicek/online-marketing/>.
- [29] E-shop z pohledu zákazníka | interval.cz. *Interval.cz* [online]. 2006 [cit. 2019-05-06]. Dostupné z: <https://www.interval.cz/clanky/e-shop-z-pohledu-zakaznika/>.
- [30] Radovan Švadlenka, Libor a Madleňák. *Elektronické obchodování*. Pardubice: Institut Jana Pernera, 1. vyd. edition, 2007.
- [31] Eshop: testováno na lidech | daniel nytra. *Danielnytra.cz* [online]. 2014 [cit. 2019-04-09]. Dostupné z: <https://www.danielnytra.cz/eshop-testovano-na-lidech/>.
- [32] Důvěryhodnost a bezpečnost e-shopu, aneb šifra mistra https | blog eshop-rychle.cz. *Blog.eshop-rychle.cz* [online]. [cit. 2019-04-14]. Dostupné z: <https://blog.eshop-rychle.cz/duveryhodnost-bezpecnost-eshopu-https/>.
- [33] 6 úprav, kterými zvýšíte důvěryhodnost e-shopu - radeksikyr.cz. *Radeksikyr.cz* [online]. 2016 [cit. 2019-04-09]. Dostupné z: <https://www.radeksikyr.cz/blog/6-uprav-kterymi-zvysite-duveryhodnost-e-shopu/>.
- [34] Html, hypertext markup language, html 5, wysiwyg, struktura html, sgml, world wide web, internet- slovník pojmů - stargen | webdesign. *Stargen.cz* [online]. 2009 [cit. 2019-04-14]. Dostupné z: <https://www.stargen.cz/slovník/HTML/>.
- [35] Co je css. *Pestujemeweb.cz* [online]. 2009 [cit. 2019-04-24]. Dostupné z: <http://www.pestujemeweb.cz/obsah/css/co-je-css.php>.
- [36] Less: css preprocesor pro pohodlnější stylování. *Itnetwork.cz* [online]. 2016 [cit. 2019-04-26]. Dostupné z: <https://www.itnetwork.cz/html-css/webdesign/less-css-preprocesor>.
- [37] Css preprocesory: méně psaní, vyšší efektivita - zdroják. *Zdrojak.cz* [online]. 2012 [cit. 2019-04-11]. Dostupné z: <https://www.zdrojak.cz/clanky/css-preprocesory-mene-psani-vyssi-efektivita/>.
- [38] Php /základy/. *Tvorba-webu.cz* [online]. 2006 [cit. 2019-05-03]. Dostupné z: <https://www.tvorba-webu.cz/php/>.

- [39] Callum Hopkins. *PHP okamžitě*. Computer Press, Brno, 1. vyd. edition, 2014.
- [40] Jan Voráček. Skriptovací jazyky pro tvorbu webových aplikací, 2013.
- [41] Programovací jazyky. *Programování* [online]. 2015 [cit. 2016-04-01]. Dostupné z: <http://k-prog.wz.cz/progjaz/>.
- [42] Programovací paradigma. *Naprogramujmi.cz* [online]. Brno-sever: MEDIACENTRUM GROUP, 2013 [cit. 2016-04-01]. Dostupné z: <http://naprogramujmi.cz/programovaci-paradigma/>.
- [43] Introduction - javascript | mdn. *Developer.mozilla.org* [online]. 2015 [cit. 2019-04-09]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction>.
- [44] Javascript. *Tvorba-webu.cz* [online]. 2006 [cit. 2019-04-06]. Dostupné z: <https://www.tvorba-webu.cz/javascript/>.
- [45] Jiří Kosek. *PHP a XML*. Profesional. Grada, Praha, 1. vyd. edition, 2009.
- [46] Lekce 1 - úvod do objektově orientovaného programování v php. *Itnetwork.cz* [online]. 2015 [cit. 2019-04-26]. Dostupné z: <https://www.itnetwork.cz/php/oop/php-tutorial-uvod-do-objektove-orientovaneho-programovani>.
- [47] Michael Kofler. *PHP 5 a MySQL 5 : průvodce webového programátora*. Computer Press, Brno, vyd. 1. edition, 2007.
- [48] Oop v php. *Programujte.com* [online]. 2011 [cit. 2019-04-08]. Dostupné z: <http://programujte.com/clanek/2009113001-oop-v-php/>.
- [49] Lekce 2 - první objektová aplikace v php. *Itnetwork.cz* [online]. 2015 [cit. 2019-04-15]. Dostupné z: <https://www.itnetwork.cz/php/oop/php-tutorial-prvni-objektova-aplikace>.
- [50] Jakub Vrána. *1001 tipů a triků pro PHP*. Computer Press, Brno, vyd. 1. edition, 2010.
- [51] Co je to framework? - webárna. *Langi.cz* [online]. 2012 [cit. 2019-05-02]. Dostupné z: <http://langi.cz/webarna/co-je-to-framework>.
- [52] Lekce 1 - úvod do nette frameworku pro php. *Itnetwork.cz* [online]. [cit. 2019-04-11]. Dostupné z: <https://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette/>.
- [53] Mvc architektura. *Itnetwork.cz* [online]. [cit. 2019-05-05]. Dostupné z: <https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor/>.
- [54] Is it mvp vs. mvc vs. mvvm that creates an odyssey for android app development? *Konstantinfo.com* [online]. [cit. 2019-04-07]. Dostupné z: <https://www.konstantinfo.com/blog/mvp-vs-mvc-vs-mvvm/>.

- [55] Mvc aplikace & presentery | nette framework. *Doc.nette.org* [online]. [cit. 2019-05-04]. Dostupné z: <https://doc.nette.org/cs/2.4/presenters>.
- [56] Úvod do mvp na androidu | ackee blog. *Ackee.cz* [online]. 2016 [cit. 2019-04-15]. Dostupné z: <https://www.ackee.cz/blog/uvod-do-mvp-na-androidu/>.
- [57] Databáze. *Databaze.chytrak.cz* [online]. 2011 [cit. 2019-04-08]. Dostupné z: <http://www.databaze.chytrak.cz/>.
- [58] Martin Horák. Návrh informačního systému, 2015.
- [59] Lehký úvod – teorie databází. *Dotnetportal.cz* [online]. 2014 [cit. 2019-04-10]. Dostupné z: <https://www.dotnetportal.cz/clanek/60/Lehky-uvod-teorie-databazi>.
- [60] Co je to relační databáze? - it slovník. *It-slovník.cz* [online]. [cit. 2019-04-28]. Dostupné z: <https://it-slovník.cz/pojem/relacni-databaze>.
- [61] Andrea Tarr. *PHP and MySQL : 24-hour trainer*. Wiley, Indianapolis, 2012.
- [62] Git - správa verzí. *Git-scm.com* [online]. 2014 [cit. 2019-04-11]. Dostupné z: <https://git-scm.com/book/cs/v1/>
- [63] Introduction - composer. *Getcomposer.org* [online]. 2012 [cit. 2019-04-06]. Dostupné z: <https://getcomposer.org/doc/00-intro.md>.
- [64] Composer: jak na různé varianty instalace » php-fashion. *Php-fashion.com* [online]. 2014 [cit. 2019-05-04]. Dostupné z: <https://php-fashion.com/composer-jak-na-ruzne-varianty-instalace>.
- [65] Composer. *It-network.cz* [online]. [cit. 2019-04-26]. Dostupné z: <https://www.it-network.cz/php/ostatni/composer>.
- [66] Handling private packages - composer. *Getcomposer.org* [online]. 2012 [cit. 2019-04-28]. Dostupné z: <https://getcomposer.org/doc/articles/handling-private-packages-with-satis.md>.
- [67] Grafické podklady | nette framework. *Nette.org* [online]. 2014 [cit. 2019-04-09]. Dostupné z: <https://nette.org/cs/logo>.
- [68] Rychlý a pohodlný vývoj webových aplikací v php | nette framework. *Nette.org* [online]. 2010 [cit. 2019-04-06]. Dostupné z: <https://nette.org/cs/>.
- [69] Seznámení s nette frameworkem | nette framework. *Doc.nette.org* [online]. 2016 [cit. 2019-04-23]. Dostupné z: <https://doc.nette.org/cs/2.4/getting-started>.
- [70] Github - nittro/nittro: javascript client-side library to use with the nette framework. *Github.com* [online]. 2018 [cit. 2019-05-01]. Dostupné z: <https://github.com/nittro/nittro>.

- [71] Clade-is - institut biostatistiky a analýz, s.r.o. *Biostatistika.cz* [online]. [cit. 2019-04-11]. Dostupné z: <https://www.biostatistika.cz/index.php?pg=produkty-clade-is>.
- [72] Getting started with doctrine - object relational mapper (orm) - doctrine. *Doctrine-project.org* [online]. [cit. 2019-04-29]. Dostupné z: <https://www.doctrine-project.org/projects/doctrine-orm/en/2.6/tutorials/getting-started.html>.
- [73] Doctrine 2: základní definice entit - zdroják. *Zdrojak.cz* [online]. 2012 [cit. 2019-04-25]. Dostupné z: <https://www.zdrojak.cz/clanky/doctrine-2-zakladni-definice-entit/>.
- [74] Open source search & analytics · elasticsearch | elastic.
- [75] Getting started | elasticsearch reference [7.0] | elastic. *Elastic.co* [online]. 2015 [cit. 2019-04-10]. Dostupné z: <https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html>.
- [76] Ib (integration broker) - gartner it glossary. *Gartner.com* [online]. 2012 [cit. 2019-04-16]. Dostupné z: <https://www.gartner.com/it-glossary/ib-integration-broker>.
- [77] Dabbling around rabbit mq persistence, durability & message routing. *Medium.com* [online]. [cit. 2019-04-05]. Dostupné z: <https://medium.com/@kousiknath/dabbling-around-rabbit-mq-persistence-durability-message-routing-f4efc696098c>.
- [78] Messaging that just works — rabbitmq. *Rabbitmq.com* [online]. 2007 [cit. 2019-04-06]. Dostupné z: <https://www.rabbitmq.com/>.
- [79] Co je virtualizace – definice | microsoft azure. *Azure.microsoft.com* [online]. [cit. 2019-05-03]. Dostupné z: <https://azure.microsoft.com/cs-cz/overview/what-is-virtualization/>.
- [80] Introduction - vagrant by hashicorp. *Vagrantup.com* [online]. [cit. 2019-04-08]. Dostupné z: <https://www.vagrantup.com/intro/>.
- [81] Laravel homestead - laravel - the php framework for web artisans. *Laravel.com* [online]. 2019 [cit. 2019-05-02]. Dostupné z: <https://laravel.com/docs/5.8/homestead>.
- [82] Static code analysis - owasp. *Owasp.org* [online]. 2008 [cit. 2019-04-16]. Dostupné z: https://www.owasp.org/index.php/Static_Code_Analysis.
- [83] What is cyclomatic complexity? and how to calculate it | perforce. *Perforce.com* [online]. [cit. 2019-04-14]. Dostupné z: <https://www.perforce.com/blog/qac/what-cyclomatic-complexity>.

- [84] Hana Hladíková ičo: 68737424, lavičky, výpis z rejstříku (06.05.2019) | euro.cz. *Firmy.euro.cz* [online]. [cit. 2019-04-13]. Dostupné z: <https://firmy.euro.cz/subjekt-hana-hladikova-68737424#res>.
- [85] Jaké řešení eshopu si vybrat? | souki.cz. *Souki.cz* [online]. 2009 [cit. 2019-04-06]. Dostupné z: <https://www.souki.cz/jak-reseni-eshopu-si-vybrat>.
- [86] 11 best open source and free ecommerce platforms for 2019. *Ecommerce-platforms.com* [online]. [cit. 2019-05-05]. Dostupné z: <https://ecommerce-platforms.com/articles/open-source-ecommerce-platforms>.
- [87] Pokud neplánujete další alzu, může se pro začátek vyplatit pronájem e-shopu - podnikatel.cz. *Podnikatel.cz* [online]. 2016 [cit. 2019-05-04]. Dostupné z: <https://www.podnikatel.cz/clanky/pokud-neplanujete-dalsi-alzu-muze-se-pro-zacatek-hodit-pronajem-e-shopu/>.

Seznam obrázků

2.1	Části elektronického podnikání	18
2.2	Znázornění architektury MVC	32
2.3	Znázornění architektury MVP	34
2.4	Logo Nette Framework	38
2.5	Logo Doctrine	39
2.6	Logo Elasticsearch	40
2.7	Logo RabbitMQ	41
3.1	Logo HanaBrand používané převážně na výrobcích	45
3.2	Zjednodušená varianta loga HanaBrand	45
3.3	Poměr obrátu v jednotlivých kategoriích na e-shopu v roce 2018	46
3.4	Přihlášení do administračního rozhraní e-shopu	49
3.5	Přehled výpisu produktů v administraci	50
3.6	Administrace produktu	50
3.7	Přehled objednávek v administraci	51
3.8	Detail objednávky	53
3.9	Formulář pro přidání nového slevového kupónu	54
3.10	Výsledek statické analýzy kódu současného řešení	56
4.1	Logo systému Jarvis	65
4.2	Satis konfigurace	66
4.3	Ukázka zdrojového kódu s anotačním zápisem pro Doctrine 2	67
4.4	Rozdělení systému na balíčky – jednotlivé repositáře	69
4.5	Logo Jarvis App	69
4.6	Logo Jarvis Core	70
4.7	Logo Jarvis CMS	70
4.8	Logo Jarvis Shop	71
4.9	3 Moduly systému	71
4.10	Základní adresářová struktura aplikace	73
4.11	Vývojový diagram procesu asynchronního generování obrázků	79
4.12	Ukázka automaticky vygenerovaného formuláře pro úpravu měny	80
4.13	Export entity do Elasticsearch	81
4.14	Import entity z Elasticsearch	82
4.15	Formulář úpravy článku s překladem	83
4.16	Konfigurace cen produktu	84
4.17	Přihlášení do administračního rozhraní systému	86

4.18	Přihlášení do administračního rozhraní systému po změně obrázku . .	86
4.19	Nástěnka administračního rozhraní systému	87
4.20	Nástěnka administračního rozhraní systému	88
4.21	Ukázka výpisu definovaných uživatelských parametrů	89
4.22	Ukázka výběru varianty produktu ve výchozí šabloně detailu produktu	90
4.23	Open-source rozšíření Nette Config Finder	91
4.24	Ukázka výchozí šablony výpisu produktů	92
4.25	Modul s informací o plánované inventuře	93
4.26	Hlavička webu	94
4.27	Produktová prezentace na úvodní stránce	94
4.28	Produktová prezentace nejoblíbenějších produktů	95
4.29	Výpis produktových kategorií	95
4.30	Výpis vybraných referencí	96
4.31	Produktová prezentace oblíbených produktů	97
4.32	Patička webu	97
4.33	Hlavička kategorie produktů	98
4.34	Výpis produktové kategorie s filtry a podkategoriemi	99
4.35	Hlavní informace o produktu	100
4.36	Výpis souvisejících produktů	100
4.37	Historie změn v repositářích systému	101
4.38	Statická analýza zdrojových kódů systému Jarvis	102
4.39	4 Moduly systému	104